SOME EXPERIMENTS WITH GIVING A COMPUTER PROGRAM THE ABILITY TO LEARN TO PLAY A SIMPLE GAME BY ASKING ADVICE OF A HUMAN TEACHER

by

Paul W. Rautenbach B.Sc.

- o O o -

A thesis submitted for the degree of

Doctor of Philisophy

- o O o -

The Department of Electrical Engineering and Electronics

Brunel University

Uxbridge

Middlesex

December 1982

ABSTRACT

A model of learning in a simple game playing context is simulated with the aim of gaining insight into a pattern learning process rather than the subtleties of the particular game.

The simulation consists of a model of a learning player able to ask help of a (human) teacher when unable to make its own decisions on the basis of previously learned knowledge. An opponent is provided by either a human player or another example of a simulated player.

To begin with, the learning player is new to the game and asks many questions of its teacher. As its playing experience increases, so it relies less on external advice, making more of its own decisions based on advice given in similar game situations encountered previously.

The system can be set up to allow fully automated trials between simulated players with varying learning experience in order to compare their performance.

The difficulty of conveying to the learning model the intention of the teacher's advice, even in the context of a simple game, highlights more general problems in Artificial Intelligence. The problem is a fundamental difference in the mechanisms of human and computer activity.

Having done this work it is possible to conclude that the world of a computer, and the natural world, impose such different constraints that they encourage the development of very different mechanisms.

ACKNOWLEDGEMENTS

First I must thank Professor Igor Aleksander for all the help he has given while supervising this thesis and the faith he has shown in me in waiting the many years it has taken to complete.

I am grateful to the Science Research Council for their grant which enabled the research to be done and to the Department of Electrical Engineering and Electronics of Brunel University for providing the research facilities.

Also, I would like to acknowledge the contribution of various colleagues who have shown interest in my work. Their insistence on being told what my thesis is about, more than anything, has clarified my thinking.

Special thanks are due to Mrs. Eileen Tedman for her persistence in typing and changing the countless drafts of this thesis.

Finally, I would like to thank David Wright for giving the assistance of the Microprocessor Technology Centre at Standard Telecommunication Laboratories Limited in the preparation of the thesis.

CONTENTS

Page

1. INTRODUCTION 1

1.1 Statement of the Problem 1

1.2 Related Work 2

1.2.1 Game-Playing Programs 2

1.2.1.1 Historical Summary 5

1.2.2 Artificial Intelligence 7

1.2.2.1 Bridging the Gap 9

1.3 Thesis Outline 11

2. THE GAME-PLAYING ENVIRONMENT AND ITS IMPLEMENTATION 14

2.1 The Kenju Game 14

2.1.1 Kenju and Other Similar Games 14

2.1.2 Discussion of Kenju 15

2.2 Software to Support the Game 19

2.2.1 Overview of Software Implementation 19

2.2.2 Human-Computer Interaction 19

2.2.2.1 Command Mode 21

2.2.2.2 Game-Playing Mode 21

2.2.3 Board-State Representation 26

2.2.3.1 The Line-List 27

2.2.3.2 The Place-Table 27

2.2.3.3 Combined Line-List/Place-Table Structure 31

2.3 Discussion 31

3. COMPUTER PLAYER SIMULATIONS AND LEARNING METHODS 32

3.1 Introduction 32

3.2 System 1 33

3.2.1 Overview 33

3.2.2 Useful-Moves Selection 36

3.2.3 Move Situation Coder 36

3.2.4 The Learning Net 40

3.2.4.1 Single Element Nets 40

3.2.4.2 Multi Element Nets and Generalisation 43

3.2.4.3 The Net Used 44

3.2.5 Move Response 46

3.2.6 Best-Move Selector 46

3.3 System 2 47

3.3.1 Introduction 47

3.3.2 Hypothesis 47

3.3.2.1 Built-In Abilities 48

3.3.2.2 Build-Up of Knowledge 48

3.3.2.3 Use of Knowledge 49

3.3.2.4 Discussion 50

3.3.3 Useful-Moves Selection 52

3.3.4 Move Situation Codes 52

3.3.5 Changes to the Learning Net 54

3.3.6 Move Response 56

3.3.7 General Comments 56

3.4 System 3 57

3.4.1 Introduction 57

3.4.2 Move Search 59

3.4.3 Line Evaluation and Move Evaluation 61

3.4.3.1 Line Evaluation Net 61

3.4.3.2 Move Evaluation 62

3.4.3.3 Move Evaluation Heuristic 64

3.4.4 The Move Search Flow Chart 66

3.4.4.1 Choice of Starting Line (Box 1) 68

3.4.4.2 Best Move Evaluation (Box 2) 69

3.4.4.3 Updating the Best-Move-To-Date (Box 3) 69

3.4.4.4 The Iteration-Patience Parameter (Box 4) 69

3.4.4.5 Dead-End Searches (Box 5) 70

3.4.4.6 Best Line Evaluation (Box 6) 71

3.4.4.7 Dead-End Patience (Box 7) 72

3.4.5 The Structure of the Search Path 72

3.5 System 4 75

3.5.1 Relation to System 3 75

3.5.2 Teaching by Preferred Moves 76

3.5.3 Summary and Discussion 77

4. LEARNING METHODS AND TRAINING 78

4.1 Introduction 78

4.2 Learning in System 1 79

4.2.1 Training 79

4.2.2 Deficiencies 79

4.2.3 The "Pattern-Background" Problem 81

4.3 Learning Modes 84

4.3.1 Tutor Mode 84

4.4 Learning in System 2 85

4.4.1 Pattern-Background Attention 86

4.4.2 Discussion 87

4.5 Learning in System 3 87

4.6 Training the Line Evaluation Net 89

4.7 Analysis of the Line Evaluation Net 91

4.7.1 Performance Graphs 92

4.7.1.1 Discussion of Performance Graphs 98

4.7.2 Line Value Tabulations 166

4.7.3 Condensed Line Value Tables 166

4.7.4 Line Rank Tables 166

4.7.5 Total-Learning Curves 167

4.7.6 Discussion of the Line Evaluation Element 169

4.8 Parameters of System 3 172

4.8.1 DEFENDING and ATTACKING Parameters 173

4.8.2 CONFIDENCE Evaluation 182

4.8.3 Setting DEFENDING and ATTACKING 182

4.8.4 MEDIOCRE and OUTSTANDING Thresholds 183

4.8.4.1 Results of MEDIOCRE and OUTSTANDING Settings 190

4.8.5 The NODEPATIENCE and MAXDEADENDS Parameters 192

4.8.6 Computer Player vs Computer Opponent Capability 193

4.9 Training Schemes 195

4.10 Learning in System 4 200

4.10.1 Introduction 200

4.10.2 Training the Line Evaluation Net 200

4.11 Some Comments on this Chapter 201

5. DISCUSSION AND CONCLUSIONS 203

5.1 Summary of Chapters 203

5.1.1 Chapter 1 203

5.1.2 Chapter 2 204

5.1.3 Chapter 3 205

5.1.4 Chapter 4 207

5.2 Analysis 210

5.2.1 Development by Evolution 211

5.2.2 Nature and Computing Compared 213

5.2.3 Motivation 217

5.3 Summing Up 221

REFERENCES 224

APPENDICES 228

A Basic Strategies of Kenju 228

B Command Mode Commands 230

C Situation Codes 232

CHAPTER 1

INTRODUCTION

1.1 Statement of the Problem

This thesis looks at some low level mechanisms involving learning in a game-playing system. The game provides the environment for a simulated model of a player who is able to observe and change this environment through playing the game. The environment as seen by the player has the ability to change "spontaneously" with the moves of an opponent. Lastly, a teacher helps (enables) the simulated player to learn to play the game.

The initial aim of this thesis was to simulate a sufficiently complex model of human activity to exhibit subtle modes of behaviour even to the extent of including emotionality. In the event, the considerable work required to simulate the basic concepts has allowed the investigation of only rather basic and crude behaviour patterns.

The purpose of this thesis is not to contribute to the art of computer game playing or to the analysis of the particular game used. Its central contribution is in explaining the requirements of a pattern learning system, and the difficulties of using this to map onto a digital computer the processes of a far more complex and vaguely understood human player. With this latter difficulty in mind, a compromise has been attempted between two divergent paths: the desire to model faithfully some aspects of human players on the one hand, and the logic (and temptation) of using methods uniquely appropriate to a digital computer

on the other. Hence, in as much as the system can be said to learn to behave intelligently, it does so inwardly in a way appropriate for a computer program but while drawing heavily outwardly on the human example.

1.2 Related Work

Work related to this thesis lies in the fields of Game-Playing and Artificial Intelligence. A brief review of these fields now follows.

1.2.1 Game-Playing Programs

Bibliographies of game-playing by computer can be found in Nilsson (1980) and Bond (1973). Most of these programs treat game-playing as no more than a well defined problem to be solved by the usual methods of computer science, e.g. efficient tree-searching strategies. These programs, and others which use learning and pattern recognition techniques are designed for maximum playing performance by implementing expert strategies - the name of the game is to win against a human player. Nilsson's (1980) book is entirely dedicated to an explanation of techniques for AI programming.

These approaches incorporate ingenious and novel programming techniques which, as Uhr (1973) points out, exhibit more the craftsmanship of the programmer than they extend our understanding of human game-playing. Even when our understanding is improved, such understanding is usually of the structure of the particular game, rather than of a player tackling a game playing situation. The pattern recognition approaches of Zobrist (1969) and Uhr (1973) are partial exceptions, as are the learning approaches of Koffman (1968) and Elcock & Murray (1967, 1968),

which are discussed below.

Uhr makes analogies between game-playing and pattern recognition: Games are designed to fit humans and are hence often represented in a spatial way. This can lead to patterns playing a central role in many games, and recognition of these patterns providing an important part in playing the game.

Zobrist uses a visual model of game-playing by applying n-tuple characterises (feature extractors in pattern recognition terms) to various abstractions of the global game state (e.g. board state). The abstractions allow recognition of patterns which would otherwise be obscured by extraneous features.

Another analogy made by Uhr is between game-playing and problem solving. He points out the graph structure common to both and the similarity of their method: In problem solving, nodes represent expressions formed from the "givens" (axioms) and transformations are represented by the arcs. Games have "givens" in the form of sets of pieces, and legal moves are operations on the givens. Board states are represented by nodes and legal moves by arcs between nodes. Thus a win is like achieving a proof of a theorem. Nilsson's (1980) book shows that this graph structure is applicable to much of Artificial Intelligences programming. The significance of these analogies is that game-playing can be elevated above the particulars of each game to a level of general techniques applicable to many games. This is a first step to understanding how people play games in general. Pitrat's (1968) program for Chess, Go-Moku, and Tic-Tac-Toe is an illstration of this view.

Learning is a useful tool for generalising: From a number of isolated examples, a learning program can deduce the factors common to all examples which lead to a win. More importantly, an unsuccessful example can refine a postulated theory of winning by clearing up ambiguities and misconceptions.

Koffman (1968) also recognises the similarity between game-playing, pattern recognition, and problem solving. In his technique the board is analysed after a win to determine the board states from which the win could be forced. These forcing states, together with the corresponding forcing moves, are described as a generalised pattern which makes reflections and other symmetries equivalent. Future moves may then take advantage of this record of forcing situations. Koffman, like Pitrat, describes a technique which can be applied to a class of games (including Qubic, Go-Moku, Hex and Bridge-it).

Elcock & Murray (1967) have a similar method for Go-Moku which they call "deductive backtrack analysis". Their method and Koffman's are further similar in being able to automatically order alternative near-forcing moves (subgoals, or preforce patterns in Koffman's terminology) according to how quickly they can lead to a win (or forcing move in the case of Koffman).

In their 1968 paper Elcock & Murray present a technique which is able to describe the essential attributes of a board state. This allows automated recognition of all board states having the described attributes without describing any particular state in all its detail (i.e. it gives criteria for recognition of such states). Thus they suggest the ability to recognise from a specification is more important

than to be able to reconstruct a specific board state. This is the essence of generalisation. Their technique is also very similar to Uhr's (1973) use of characterisers in that both are descriptions of what to look for.

1.2.1.1 Historical Summary

Much of the work on computer game-playing has concentrated on the game of Chess since in 1950 a suitable structure for a chess playing program was proposed by Shannon (1950).

Samules Checker player (1959) was a significant advance. It was able to improve its own game by learning from experience (against an opponent) which of a set of pre-programmed heuristics it would be best to use, and the weight to be given to each.

Greenblatt's Chess program (1967) used many pre-programmed heuristics tailored to the game of Chess and was the first program able to play acceptable chess.

A more general set of heuristics allowed Pitrat's (1968) program to play Chess, Go-Moku, and Tic-Tac-Toe.

Koffman's (1968) program was able to 'discover' for itself the forcing patterns for a class of board games including Qubic, Go-Moku, and Hex. It was unusual in not using lookahead or prescribed heuristics particular to each game. Instead it relied on analysis of the moves which lead to actual wins.

Elcock & Murray (1967, 1968) described a similar technique for Go-Moku.

Zobrist's (1969) program to play GO made an interesting analogy between playing GO and visual organisation in general. Here a prerequisite to choosing a move was to calculate perceptual groupings and segmentations of the pieces which characterised the size and density of colour of each segment.

Uhr's book (1973) viewed game-playing as pattern recognition plus problem solving, with the possibility of incorporating learning.

Many of the programs referenced above model game-playing in terms of a graph whose nodes are possible states of the game, connected to form a tree of possibilities by arcs representing legal moves. Various techniques of "tree pruning" (discarding unsuccessful branches) have been widely employed (Slagle 1969), often with the "minimax" search method where a player's best move is found by looking ahead a number of moves in the tree, maximising his gains while assuming his opponent will do his best to minimise them.

A fuller survey of computer game-playing can be found on Slagle (1971) and others are suggested in Bond (1973).

Since the work of this thesis was done the field of game-playing programs has opened up with the availability of personal computers. For example, it is now possible to play against sophisticated chess programs which run on 'toy' computers.

1.2.2 Artificial Intelligence

The term "Artificial Intelligence" spans a wide spectrum from low-level work on modelling neural nets to knowledge-based work at the other extreme. It is the opinion of the author that the best work sees Artificial Intelligence as theoretical Psycology, i.e. it tries to understand natural intelligence.

The elementary functional units postulated in neural net models have little to say about the observed behaviour of human beings; investigations at the high-level end of the spectrum often have the stated (or unstated) aim of suppporting behaviour which would be identified as intelligent if exhibited by a human being, with little regard for similarity of mechanism with the human example.

In short, there is no continuity between the extremes of Artificial Intelligence - each is unconcerned with the central themes of the other. There is a lack of concern for explaining intelligent behaviour in low-level work, and little attempt to justify mechanisms in terms of human development in the high-level work.

The most extreme example of the knowledge based approach is probably Michie's (1980) work on "knowledge refining". In this work the knowledge of an expert in a particular field is entered into some form of computer database with the hope of making such knowledge more widely available. The "crude" knowledge is input from the expert into the computer where it can be "refined" by the application of various theorem provers and other software. The result is a precise and disambiguated version of the crude input, in a form applicable to decision making, (i.e. by

rule-based deductions).

The purpose of the process is to extract the refined knowledge - any understanding of the expert's own mechanism which generated such knowledge is refined out. The system is not designed to model the mechanism of building up expertise.

A less extreme and more typical example of knowledge-based work can be found in the work of Colby based on Abelson's structural theory of belief, which is itself an extension of Schank's Conceptual Dependancy analysis (in Boden 1977 and Schank & Colby 1973).

Colby's simulation of a neurotic is a model of a woman undergoing psychoanalysis, which he uses to study defensive habits. Beliefs in the model carry a "charge" which can inhibit any expression of the belief if sufficiently large. The system selects a pool of dependent beliefs called a "complex". Conflict between beliefs in a complex requires expression of the conflicting beliefs in order to dissipate their charge. This can be done by a defensive technique whereby the belief is transformed into a distorted version which has a lesser charge and can thus be expressed. (This transformation technique is similar to the technique used in the General Problem Solver (GPS) (Feigenbaum et al 1963): The difference between what you have and what you want is reduced by applying selected transforms known to lessen particular differences).

If no transformation of a conflicting belief reduces the charge to a level at which it can be expressed, the system attempts to express another belief in the complex. Eventually either a belief is expressed, or the entire complex is suppressed and the general level of tension is

raised by an increase of charge on all beliefs.

Boden's book (Boden 1977) gives other examples of knowledge based work including an extensive bibliography.

The mechanisms chosen by Colby to support his simulations of neurotic behaviour are the semantic nets of Schank. These provide a mechanism for relating words to each other by building structure around and between the meanings of the words. This structure represents the purpose behind the use of a word in a particular semantic context. It relies on built-in knowledge of the way words are used in the English language, and is a mechanism for particularising this general knowledge to example situations, which leads to elaboration of the built-in knowledge.

Most of the work on Scene Analysis and Game Playing, done under the umbrella of Artificial Intelligence, is at the "application oriented" extreme. The understanding gained is of the application - the semantic structure of English, end-game strategies for chess, or the geometry of polyhedra.

The preoccupation of the latter fields of Artificial Intelligence with the importance of (computer) language tools to express and animate models of intelligence obscures the lack in such work of any real attempt to explain how these models might be supported on the natural computer - Man.

1.2.2.1 Bridging the Gap

The no-man's-land between the extremes of "functional bottom-up" and "behavioural top-down" is sparsely populated by a few isolated

investigations which manage to be interpretable in terms of both points of view.

The work of Andreae (1977) seems to be one such example. His choice of applications is essentially arbitrary and for illustration only; the mechanisms used are primitive and simple enough to have possibly developed (by accident) during evolution.

In essence, his model depends on the reasonable assumption that things will happen again in the context in which they have been observed to have happened before. Thus, his mechanism is a predictive sequence memory which remembers the event sequences of the past, and uses them to anticipate the future. Events are fundamentally uninterpreted symbols in the temporal pattern of history. It is this lack of interpretation which frees Andreae's work from the pre-occupations with particular applications (and their purposeful actions) which bedevils so much of Artificial Intelligence. Andreae's model is plausible because it simply reflects an observable property of nature - repetition. Any "meaning" in the sequences, patterns, and symbols, is in the eye of the beholder.

A corresponding view is to be found in Kelly's (1955) "Personal Construct Theory" (Bannister and Fransella 1971). This theory of Psychology acknowledges that all intelligence is observed through the intelligence of the observer and thus has no purely objective expression, (i.e. intelligence is in the eye of the beholder). The central theme is one of the intelligent being, man in this case, endeavouring to prove his internal understanding (personal constructs) by testing his predictions and searching out supporting evidence; and when new evidence does not support a construct it refines it by refuting misconceptions, or deciding ambiguities, leaving a sharper

understanding.

In Andrea's and Kelly's work a novel event becomes a challenge to the system to predict and force another encounter with such an event, thereby establishing the context in which it occurs. Thus a novel event can be understood and unified with existing constructs.

The fascination of intelligence is not in an ability to do, but in the method of doing. It has more to do with cleverness than with knowledge. The importance of intelligence is better captured in the "intelligent guess" than in the raw application of knowledge characterised by an "informed opinion". Access to relevant information is of secondary importance to the skill of deciding what is relevant.

The support of this intelligent method requires both semantic structures by which the supporting being can organise and relate (i.e. understand) information, and mechanisms which implement and manipulate these organisations and relations. Artificial Intelligence must be concerned with bringing these two aspects together.

1.3 Thesis Outline

This first chapter has given a background to the thesis, both in terms of the work of others and the initial aims of this work.

It has tried to indicate the author's approach to Artificial Intelligence as the study of natural intelligence, rather than the invention of intelligent artifacts. Also, the interest in game-playing is only in providing a world in which intelligent activity may be studied.

Chapter 2 introduces the game of Kenju; chosen as the game to be played by the computer simulation. The game's implementation is outlined, as is the interactive interface to it and to the game playing algorithms of Chapter 3 and 4.

Chapters 3 and 4 contain the bulk of the thesis, covering the development of four models of a learning Kenju player. A section of Chapter 3 is devoted to each of these four learning systems. Each learning algorithm is explained in terms of the mechanisms employed.

Learning system 1 gives an initial insight into the problems. System 2 is a re-working of the initial system in the light of its deficiencies. System 3 and system 4 depart from the train of thought represented in the development of system 2 from system 1. More emphasis is placed on modeling the way a human player learns and less on the computing techniques used.

The four learning models show an evolution of ideas and of learning mechanisms.

Chapter 4 looks at the four models of a Kenju player from the point of view of the learning each is able to do. The training methods used are described and results are presented for system 3. Analysis of the results leads to few clear conclusions.

Chapter 5 then summarises the thesis and tries to understand the reasons behind the inconclusive results. The suitability of using a computer to support an evolutionary approach to developing an intelligent program is questioned.

Finally, a return is made to the discussion of Artificial Intelligence introduced in Chapter 1. In particular, a source of motivation and goals is suggested.

CHAPTER 2

THE GAME-PLAYING ENVIRONMENT AND ITS IMPLEMENTATION

2.1 The Kenju Game

Chapter 2 explains the game used as the environment in the simulation and describes the computer support required for its implementation.

2.1.1 Kenju and other similar Games

A simplified version of the game Renju, called Kenju in this thesis, is the environment of the model. Renju is a Chinese game which is best thought of as Naughts and Crosses (Tic-Tac-Toe) extended to five, rather than three in a row.

Kenju differs from Naughts and Crosses in two main ways: whereas Naughts and Crosses is played on a three by three board, Kenju has, theoretically an unbounded board, and (as for Renju), a winning line requires five adjoining moves.

The form of Renju closest to Kenju is Gomoko (five stones) described in Elcock and Murray (1967, 1968).

Much of what is known about the game of Renju (Gomoko) can be applied to Kenju. Certain restrictions on board size and forbidden moves in Gomoko (especially in its more advanced forms) have been omitted in Kenju for simplicity. Uhr (1973) points out that the distribution, in terms of programming effort, of "rigidity" in a game can be avoided by a game

which minimises special cases. Kenju's lack of special boundary positions and differing types of pieces each with their own particular legal moves, gives a simple and general computer implementation. Furthermore, to some extent Kenju does exhibit a "simple, natural, regular substrate of rules or principles that account for the rules of the game" which Uhr (1973) refers to ("abstractness") as desirable for developing general principles rather than mechanisms specific to each game.

2.1.2 Description of Kenju

Kenju is played by two opposing players placing "stones" on the intersections of a boundless square grid (the board).

Each player has an unlimited supply of stones of his colour (Black or White). During his move a player must place one of his stones on a vacant grid intersection (board position). The two players, Black and White, take alternate moves until one player wins the game by completing a "winning line". A winning line must contain five adjoining stones of the winner's colour forming a straight line in one of the four directions: horizontal, vertical, positive diagonal, or negative diagonal.

For example, a winning move for Black would be as indicated by the cross in Fig. 2.1.

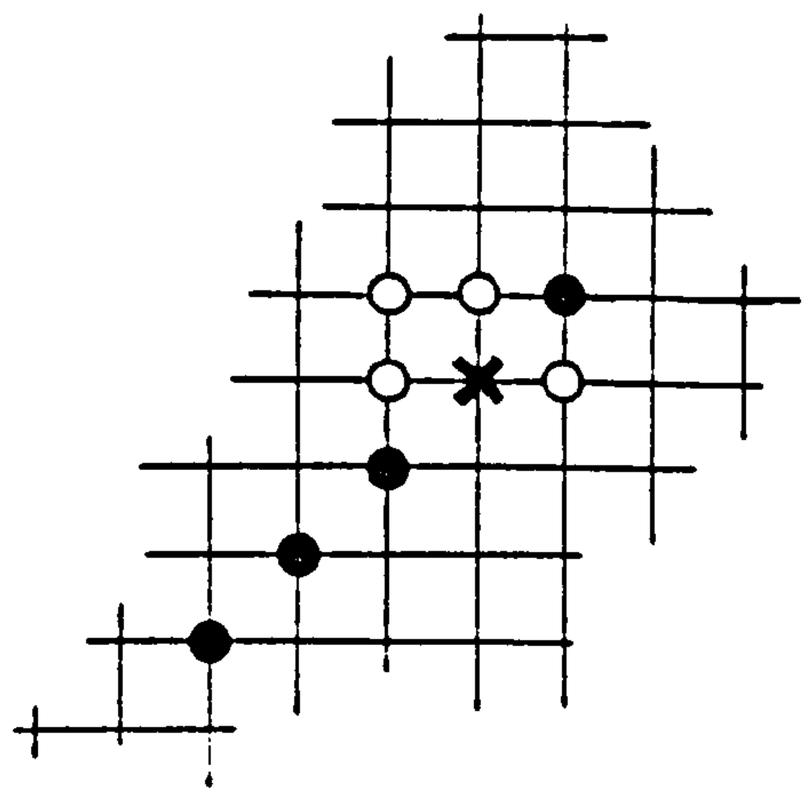


Figure 2.1 A winning Move

Playing Kenju involves the formation of certain line patterns which can lead to a winning line. These patterns, shown in Fig. 2.2 are taken from Elcock and Murray (1967). The games is won by building towards a winning line in such a way as to achieve the winning line before one's opponent completes his. Thus a player's basic strategy might be to ensure that each move taken extends "partial lines" (incomplete potentially winning lines) of his colour while blocking those of his opponent. This strategy is improved by trying to extend (and/or block) more than one partial line with a single stone placed at their intersection.

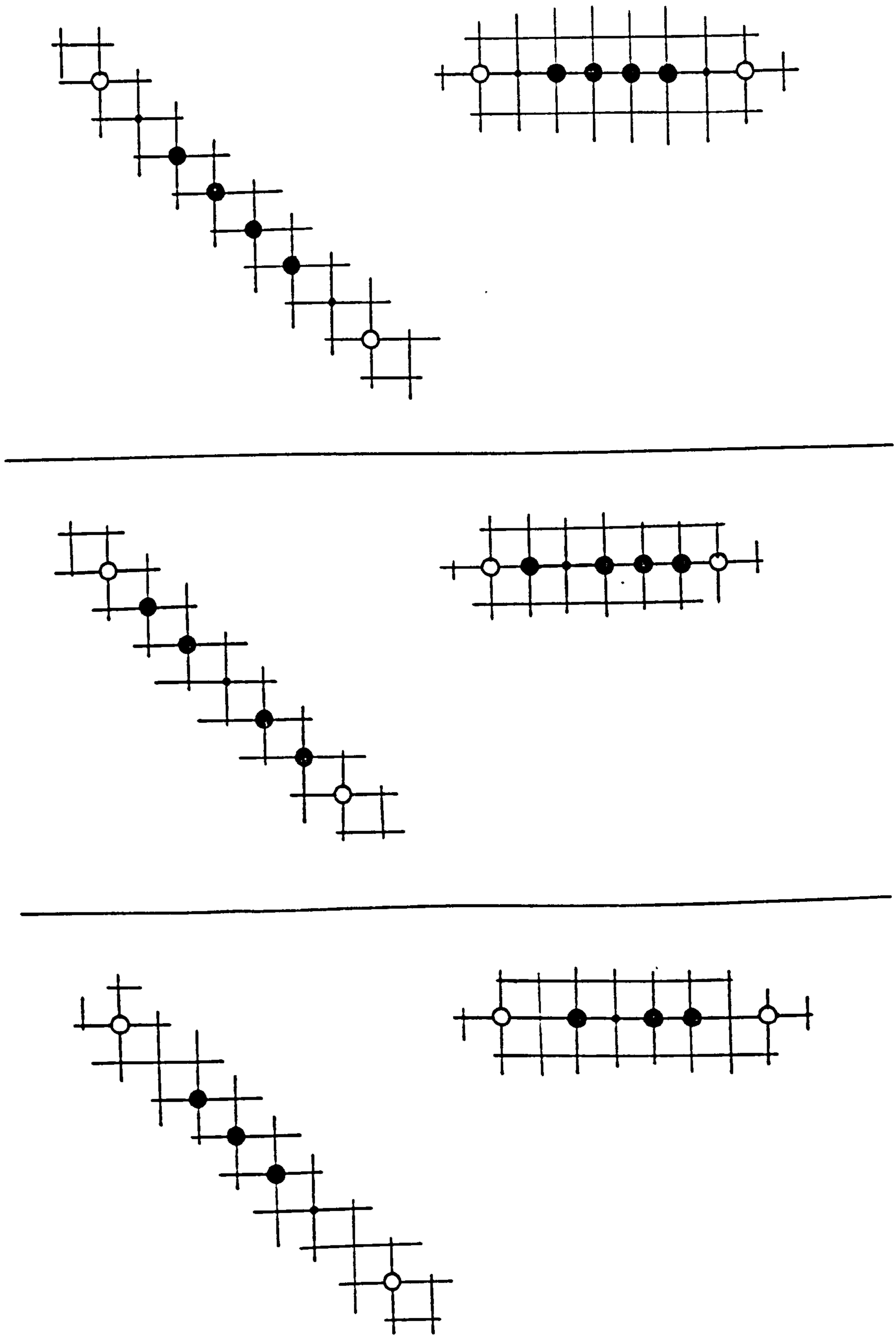


Figure 2.2a Winnable Line Patterns

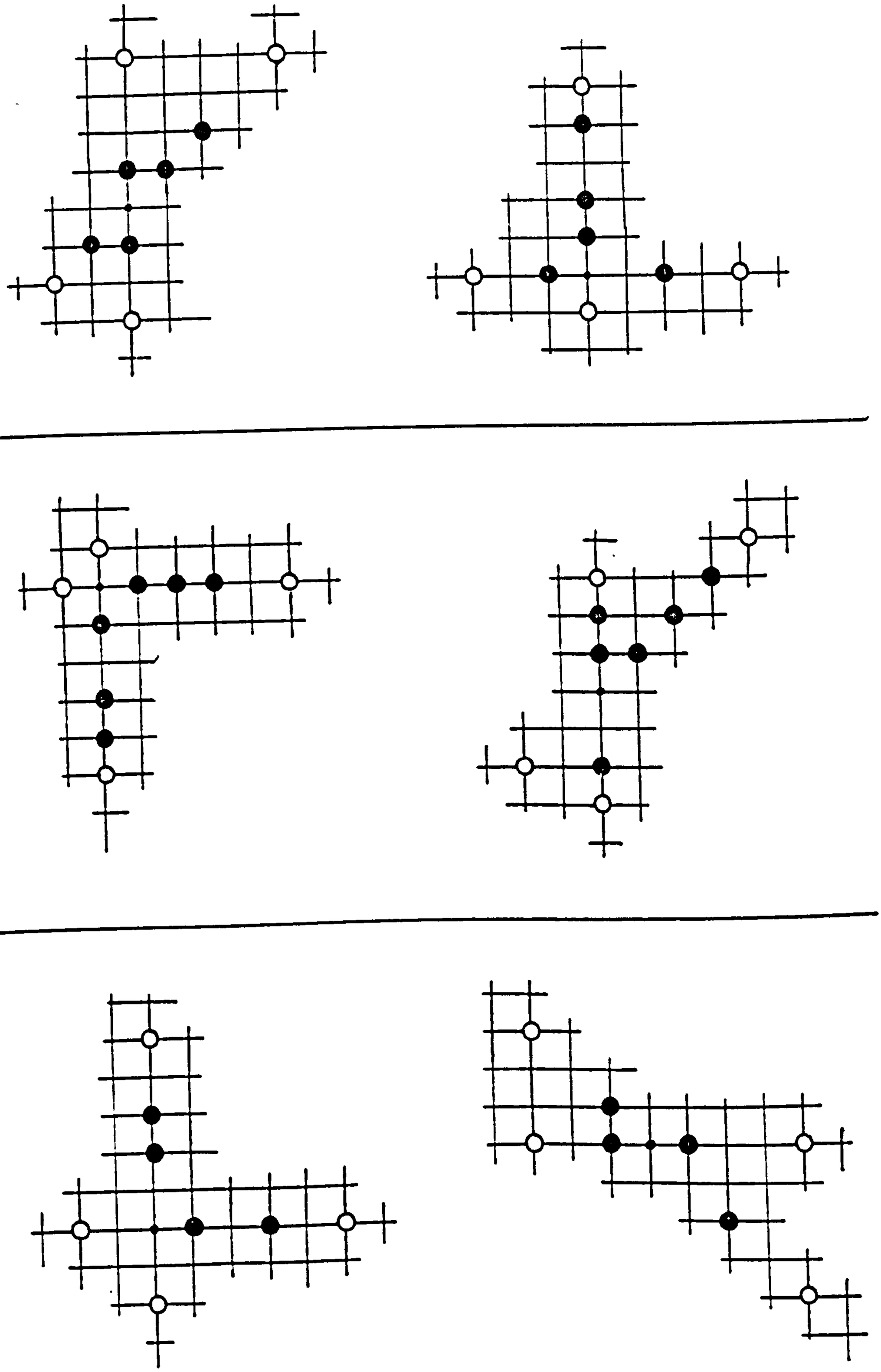


Figure 2.2b Winnable Line Patterns

A definition of full Renju with opener's restrictions can be found in Oda and Womack (1972) where the basic terms of Renju are defined. Appendix A gives the modified basic strategies of Kenju adapted from those given for Renju in Oda and Womack.

2.2 Software to Support the Game

2.2.1 Overview of Software Implementation

The simulation of the game (environment) and the computer model of a player requires the implementation of three software modules:

1. Human-Computer Interaction
2. Board State Representation
3. Computer Player Simulation

Modules 1 and 2 are described below. Module 3 is discussed in Chapters 3 and 4.

2.2.2 Human-Computer Interaction

A human player (or teacher) interacts with the computer mainly through a graphics terminal. The light-pen and keyboard allow input to the computer. The computer indicates its response through text and graphics on the graphics screen.

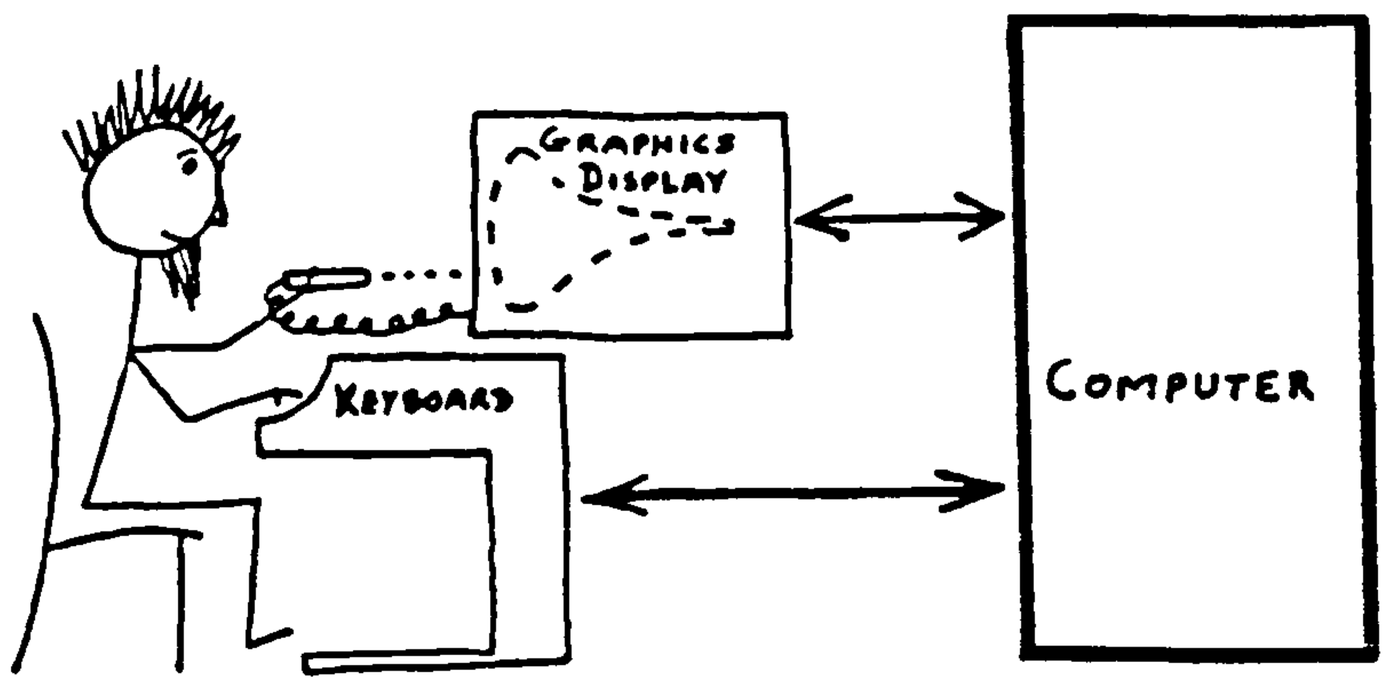


Figure 2.3 Human-Computer Interface

Two modes of interaction with the computer are required: command mode and game-playing. All initial dialogue and inspection of the internal state of the simulation takes place through the keyboard in command mode. Human player and teacher actions are in game-playing mode using the light-pen and graphics (and occasionally the keyboard). Simple methods of switching between modes are provided.

Fig. 2.4 shows the format of the graphics screen during game-playing mode. During command mode the entire screen is used for displaying text.

2.2.2.1 Command Mode

Command mode allows specification of input and output channels, inspection and initialisation of data areas and parameters, and initiation of games and tests (experimental trials). A list of commands is given in Appendix B.

Channels are provided for the input of commands, teaching advice, and initial states of parameters and data areas (computer player's experience). Internal parameters and data areas may be output as numerical and pictorial printouts. Other output channels are used for computer player dialogue, saving the computer player's experience in the form of parameter values and data structures, and for selectively tracing the path of execution of the computer player algorithm.

Command mode is entered initially when the program is entered at the start of a run. Game-playing mode is subsequently entered from command mode by requesting a game to be played.

2.2.2.2 Game-Playing Mode

Moves are taken (by a human player) using the light-pen and displayed pictorially on the graphics screen with previous moves and the computer player's response.

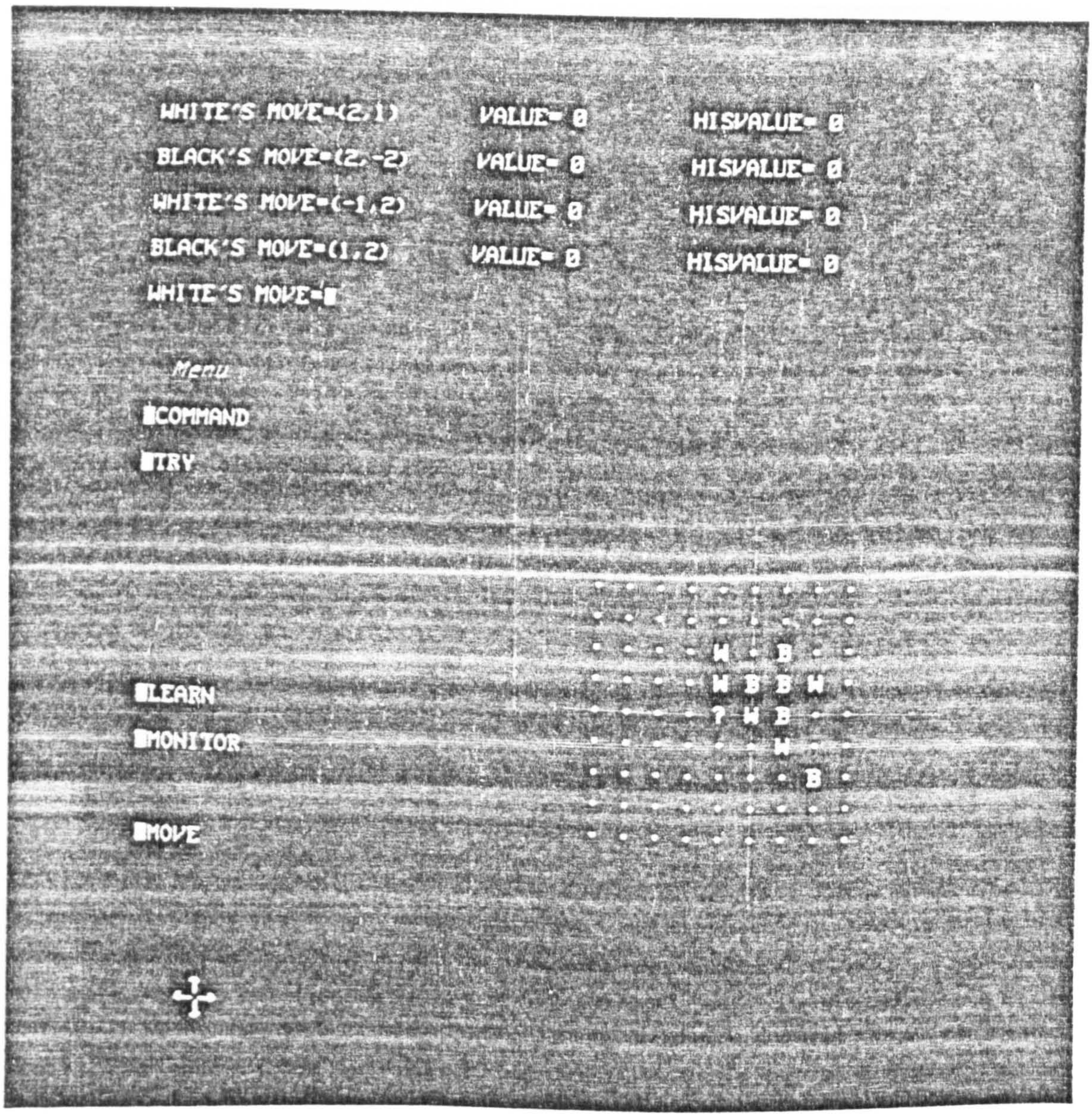


Figure 2.4a Screen Format (game-playing mode)

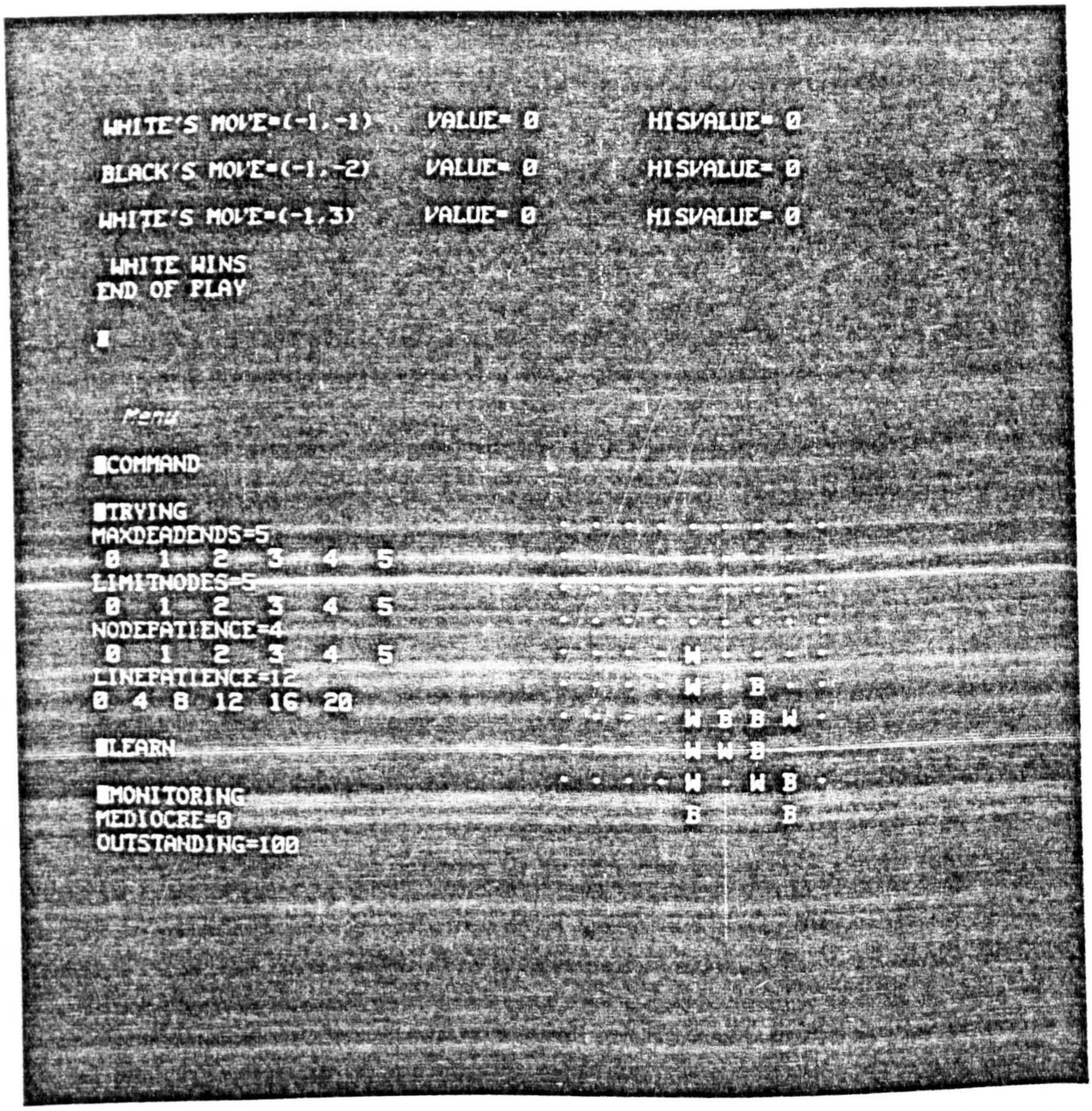


Figure 2.4b Screen Format (game-playing mode)

To take a move the human opponent points with the light-pen (moving a light-pen tracking grid) at the desired board position displayed on the screen. He must then confirm his move by selecting (using the light-pen) the "MOVE" option in the "menu" (displayed on the screen) before it is accepted. He may change his choice of move any number of times before confirming. His current choice appears on the screen as a flashing question-mark at the selected board position.

The moves of the two players are displayed on the screen as Bs and Ws for Black and White respectively, showing the current state of the game. At the end of a game the winning line is indicated by a row of flashing Bs or Ws.

Other information displayed during a game is described in Chapter 4 where teacher interaction is described.

Command mode may be re-entered from game-playing mode at any time by selecting (by light-pen) the "COMMAND" option in the "menu". A return to the point at which game-playing was left may be achieved using the "return" command from command mode.

Fig. 2.5 illustrates the possible human interactions in the form of a flow diagram.

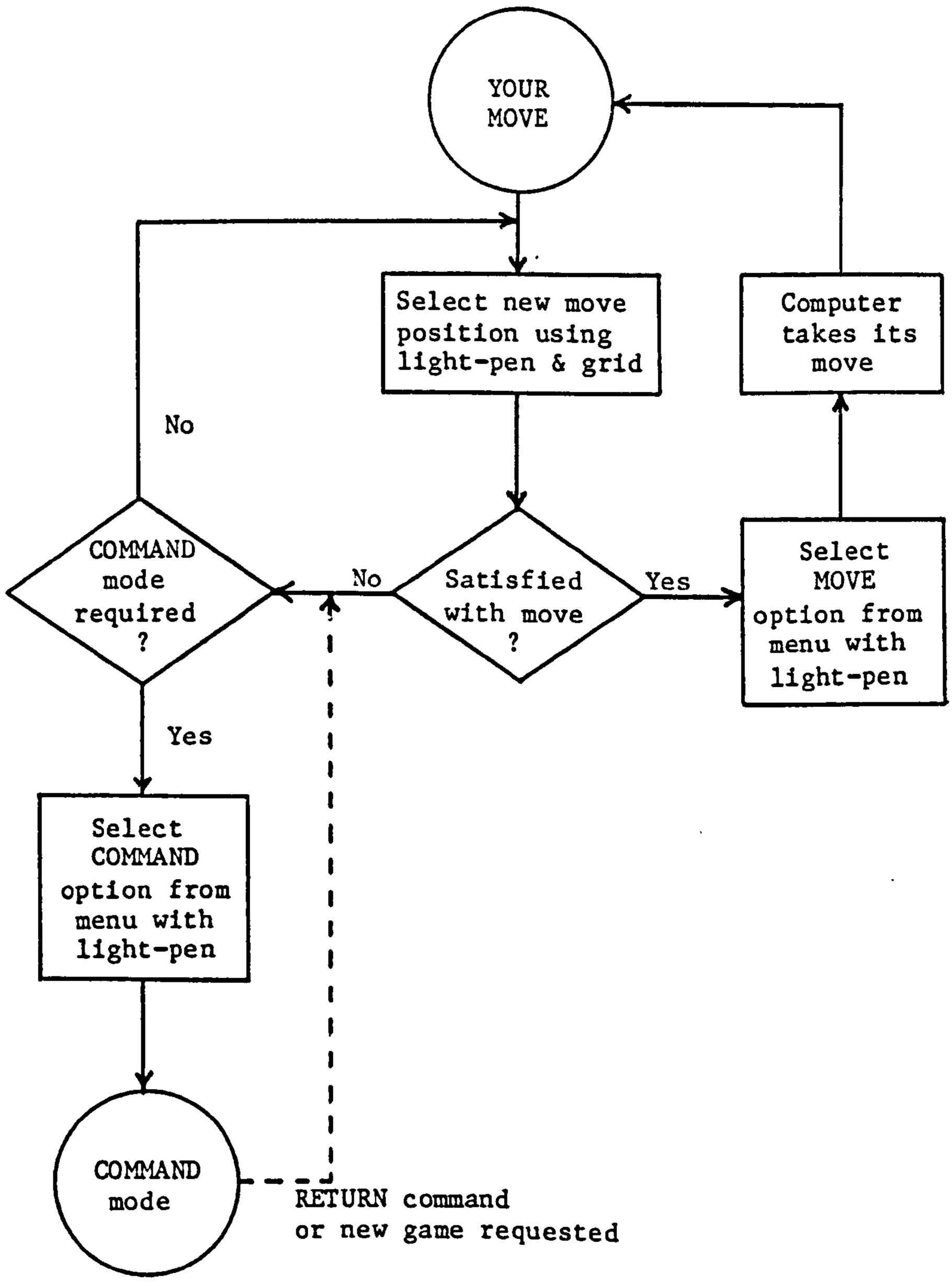


Figure 2.5 Interaction Flow Diagram.

2.2.3 Board-State Representation

The state of the board is not represented in the computer as an array of board positions because this is wasteful in terms of storage and complicated by the boundless dimensions of the board. A direct array representation would also necessitate extensive pattern recognition routines, such as the Characterisers (feature extractors) described by Uhr (1973), to isolate lines of stones.

Instead, a record of partially occupied lines of length five is maintained with a cross linked record of board positions contributing to these lines. Only those lines representing possible winning lines (lines containing stones of only one player) are retained. These "live lines" represent all the partially occupied lines which could subsequently become occupied in all five positions by one player, thus constituting a winning line. Such lines (live lines) will be referred to simply as "lines" for short, and the board positions in such lines as "places". (The representation of the board is similar to the list-structured representations suggested by Uhr (1973) for Tic-Tac-Toe (Naughts and Crosses) where the "rays" of Tic-Tac-Toe correspond to lines in Kenju).

With this reduced representation (in terms of information and storage) the board state information relevant to the playing algorithm is directly accessible as lines and positions within lines. It is also easily expandable to accommodate virtually unbounded growth of the game over the board in any direction.

Given any line, its constituent board positions (places) can be accessed. Given a particular place in a line, all lines intersecting at

that board position (i.e. containing that place) can be accessed. Thus it is possible to establish the "spheres of influence" (Uhr 1973) of lines and places, and the representation of the board is well suited to access and manipulation by the algorithms using it.

A pictorial "map" of the board can be generated from the representation by recording calculated cartesian co-ordinates (relative to the position of the first move) with each record of a board position.

The three data structures used to represent the state of the board are now described in detail, they are: the list of line records, the table of board positions (places) contributing to at least one line, and a pool of unused line records. The line-list and place-table records are cross linked to allow access by line or by place as mentioned above.

2.2.3.1 The Line-List

A variable length list of line records (line-blocks) is maintained as a linked list structure by "borrowing" and "returning" line-blocks from the list of free line-blocks. Each line-block has the structure given in Fig. 2.6.

2.2.3.2 The Place-Table

The place-table is an expanding table of vacant and occupied places contained in at least one line. Each table entry (place-block) has the form shown in Fig. 2.7.

BLOCKLINK	POSN1	POSN2	POSN3	POSN4	POSN5
LINESTATUS	XLINK1	XLINK2	XLINK3	XLINK4	XLINK5

BLOCKLINK - List link to next line-block in line-list (or in free-list if unused).

POSNn - Pointer to the place-block in place-table corresponding to position n in this line.

XLINKn - Cross link to link up lines at the position where they intersect or cross.

LINESTATUS - Line status containing two fields:

MOVES	ORIENTATION

MOVES - the number of occupied positions (moves) in this line.

ORIENTATION - the direction of this line i.e. horizontal (-), vertical (|), positive (/), or negative diagonal (\).

Figure 2.6 Line-Block Structure.

PLACESTATUS	XCO	YCO	XLIST

PLACESTATUS - status of this place with three fields:

COLOUR	MOVENUMBER	LINES

COLOUR - signifies that this place is either vacant (V), occupied by Black (B), or occupied by white (W).

MOVENUMBER - specifies the move in the game when this place became occupied.

LINES - gives the number of lines crossing at this place (i.e. the length of XLIST)

XCO - x-coordinate of the position of this place on the board (relative to the position of the first move).

YCO - y-coordinate of the position of this place on the board (relative to the position of the first move).

XLIST - points to the first XLINK element in the linked list which ties together lines intersecting at this place.

Figure 2.7 Place-Block Structure

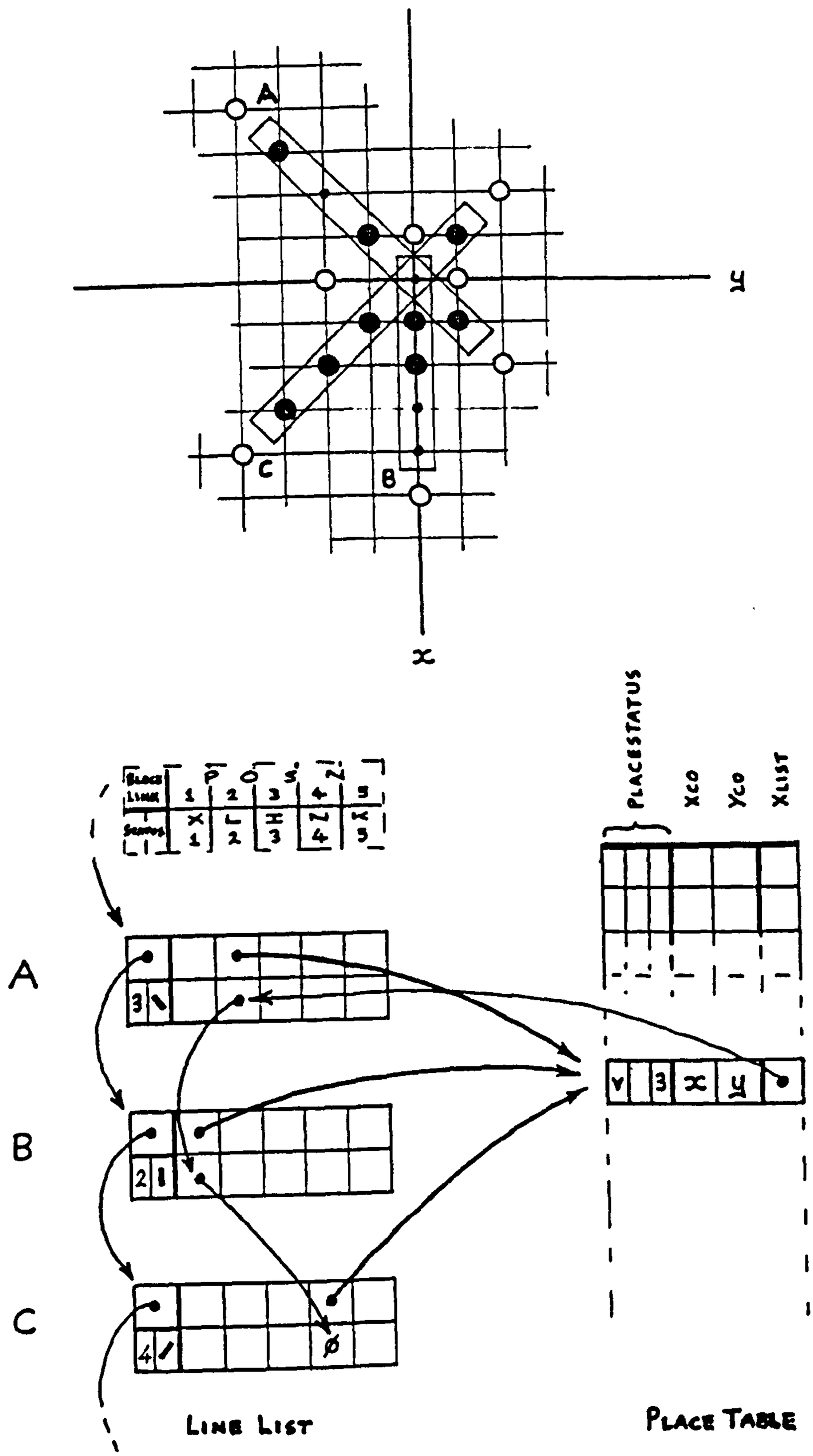


Figure 2.8 Line-List/Place-Table Combination

2.2.3.3 Combined Line-List/Place-Table Structure

Linking information associated with the crossing point, (x,y), of three lines, A, B, and C, is shown diagramatically in Fig. 2.8. The common place, (x,y), is the 2nd, 1st and 5th position in lines A, B, and C respectively.

2.3 Discussion

The computer implementation of the Kenju game as an environment for a simulated player was the first part of the model to be programmed. It has survived unchanged during various versions of the simulated player (described in Chapter 3 and 4). The robust nature of the programmes derive from the flexible, but simple, representation of the state of the board in terms of partially occupied lines rather than a two dimensional board matrix.

The representation of the board is separate from the model of a player which interacts with the board. Access to the board is through a set of routines which give the requested information about the state of the board or make the requested changes to the state of the board.

CHAPTER 3

COMPUTER PLAYER SIMULATIONS AND LEARNING MODELS

3.1 Introduction

Four learning models are described below referred to as System 1 to System 4.

System 2 is a development of the initial system, System 1. Both rely on a RAM net (see Section 3.2.4) to value possible moves. The move evaluation is based on the pattern recognition property of the net, where the pattern is a coded representation of the board positions surrounding the move.

Systems 3 and 4 represent a shift of emphasis from the move evaluation function to the chain of attention of the player across the board during his search for a move. The evaluation net of System 2 has been reduced to a simple look-up table and a heuristic routine. System 4 has the structure of System 3 but learns from moves whereas System 3 learns from lines.

Hence, learning in Systems 1 and 2 relies on recognition of the pattern of neighbouring vacant and occupied positions around a move. This constitutes a more complex learning element than the simple line pattern evaluation net or look-up table (see Section 3.4.3.1) of Systems 3 and 4 which in turn have a more sophisticated mechanism for directing the player's search for a move.

In short, Systems 1 and 2 have a simple control structure but acquire complex knowledge, and Systems 3 and 4 use simple knowledge in a more complex control structure.

The training methods, results and observations of the four systems described here in Chapter 3 are discussed later in Chapter 4.

3.2 System 1

3.2.1 Overview

This section outlines the first learning system, which gives an initial insight into the problem and suggests changes leading to the systems described in the following sections.

The selection of the computer player's next move divides into two smaller tasks. First, only that subset of all vacant board positions which interact directly with existing moves are selected for consideration as possible moves. The best of these "useful moves" is then chosen, during a second phase, by consulting "past experience".

The first step is thought of as an attention mechanism restricting the application of evaluation algorithms to useful moves. Each move which has been selected as being worth investigating is evaluated, in association with its surrounding board positions, to establish the value of its situation in the game. That is, evaluation of a move takes place in the context of the lines which contain the move, and the position it occupies in each line.

The contextual information is coded to form a bit pattern applied as address inputs to a Random Access Memory (RAM) network structured as a learning net (as explained in Section 3.2.4 and in Aleksander, Stanham and Wilson (1974)).

The response of the net to the input pattern is the trained response to (value of) the board situation in which the move is found. For example, it could be a situation in which the move would complete a winning line in one direction and should therefore have a trained response which is high.

Having acquired responses for all useful-moves the position of the move with the best response is taken for the computer's move.

The detail of the move search which follows should be read with reference to Fig. 3.1.

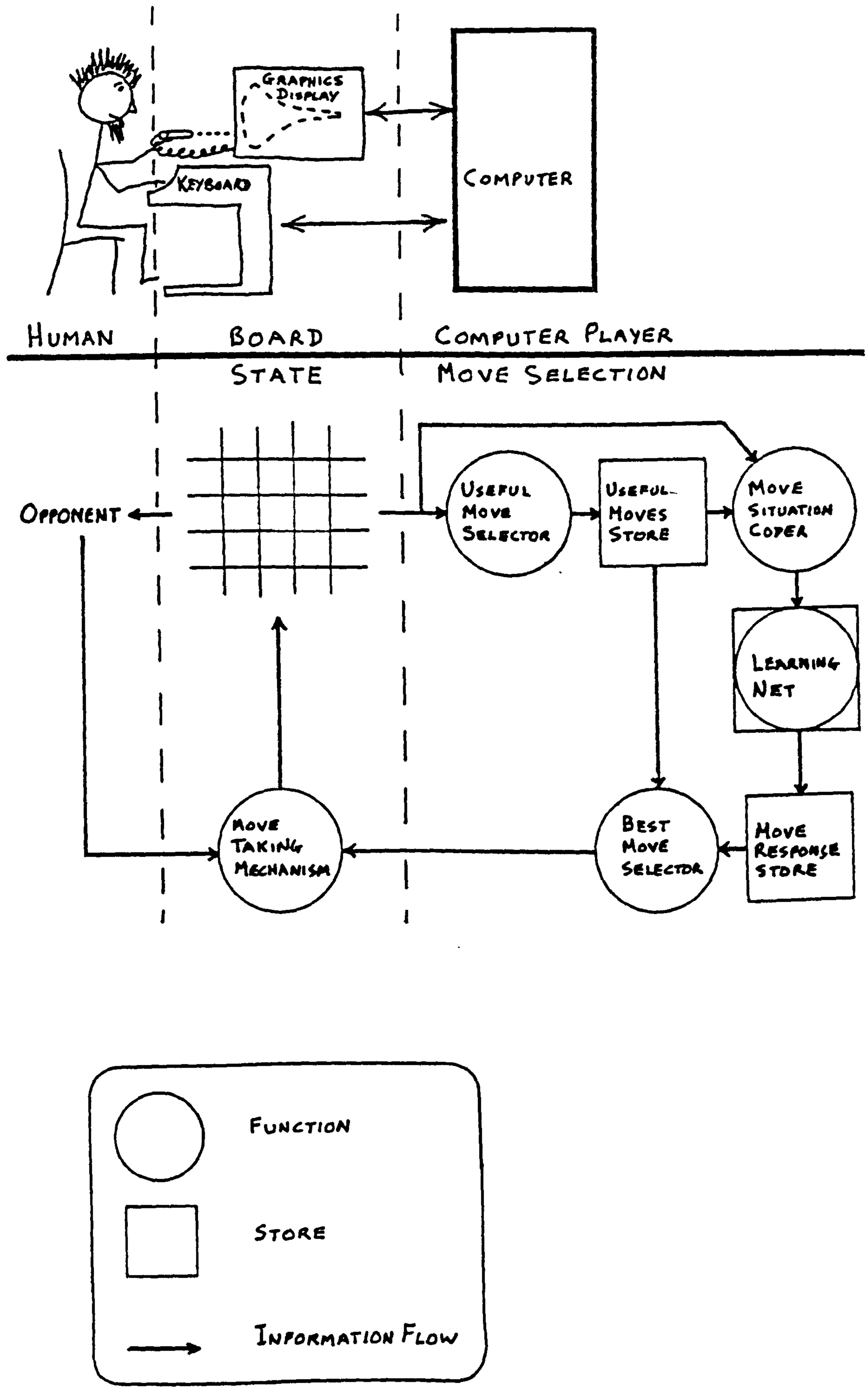


Figure 3.1 Functional Block Diagram

3.2.2 Useful-Moves Selector

The algorithm for selecting "useful moves" could be improved by using more sophisticated (but standard) pattern recognition (c.f. Uhr's (1973) characterisers, Zorbrist (1969), Koffman (1968)). Since it is not of major interest in this model the selector simply assembles all vacant places into a store for useful-moves. This effectively restricts the computer's chosen move to be one of those vacant board position which surround occupied positions and form possible winning lines with them.

In detail, the algorithm scans the place-table (see Section 2.2.3.2), selecting only vacant places for the useful-moves store.

3.2.3 Move Situation Coder

Each of the moves in the useful-moves store is associated with neighbouring positions, in those lines which it occupies, to form a code which uniquely specifies the board situation surrounding the move. The coding used is very similar to that used in Elcock and Murrary (1967).

The coding algorithm finds, for each direction and for each colour, the number of lines containing the move and having a maximum number of occupied positions. Thus, all those lines containing the maximum of occupied positions (of a particular colour) for a particular direction are counted. The count of these lines and their degree of completeness (the number of occupied positions they contain) is combined to form the context codes for each colour in each direction. Appendix C gives the correspondence between patterns and codes.

The notation for codes in each direction is:

[PL.PM/OL.OM]d or [/]d

PL refers to the number of the player's lines which contain the move position being coded, with a maximum, PM, moves of the player. OL refers to the number of the opponent's lines which contain the move position being coded, with a maximum, OM, moves of the opponent. The suffix, d, gives the direction of the lines contributing to that [/]d code: H, V, P or N, indicating Horizontal, Vertical, Positive diagonal, or Negative diagonal respectively.

The four [/]d codes are arranged, arbitrarily, in numerical order (the bit patterns of the codes viewed as binary values) to form the bit pattern of the full situation code applied to the learning net address inputs. This has the effect of generalising situations encountered in one orientation to all rotations and reflections of that orientation. Similar techniques are employed for the same purpose of generalisation by Elcock and Murray (1967, 1968), Zobrist (1969), Uhr's (1973) static similarity, and Koffman (1968).

The full code describing the board situation around the move position in all directions might be, for example:

[PL.PM/OL.OM]H [PL.PM/OL.OM]V [PL.PM/OL.OM]P [PL.PM/OL.OM]N

or []HVPN

The total code length is derived from the number of bits required to specify the PL.PM (or OL.OM) pairs. In any direction there are nine adjacent positions including the move position itself (four to each side of the move), which form lines of five with the move position.

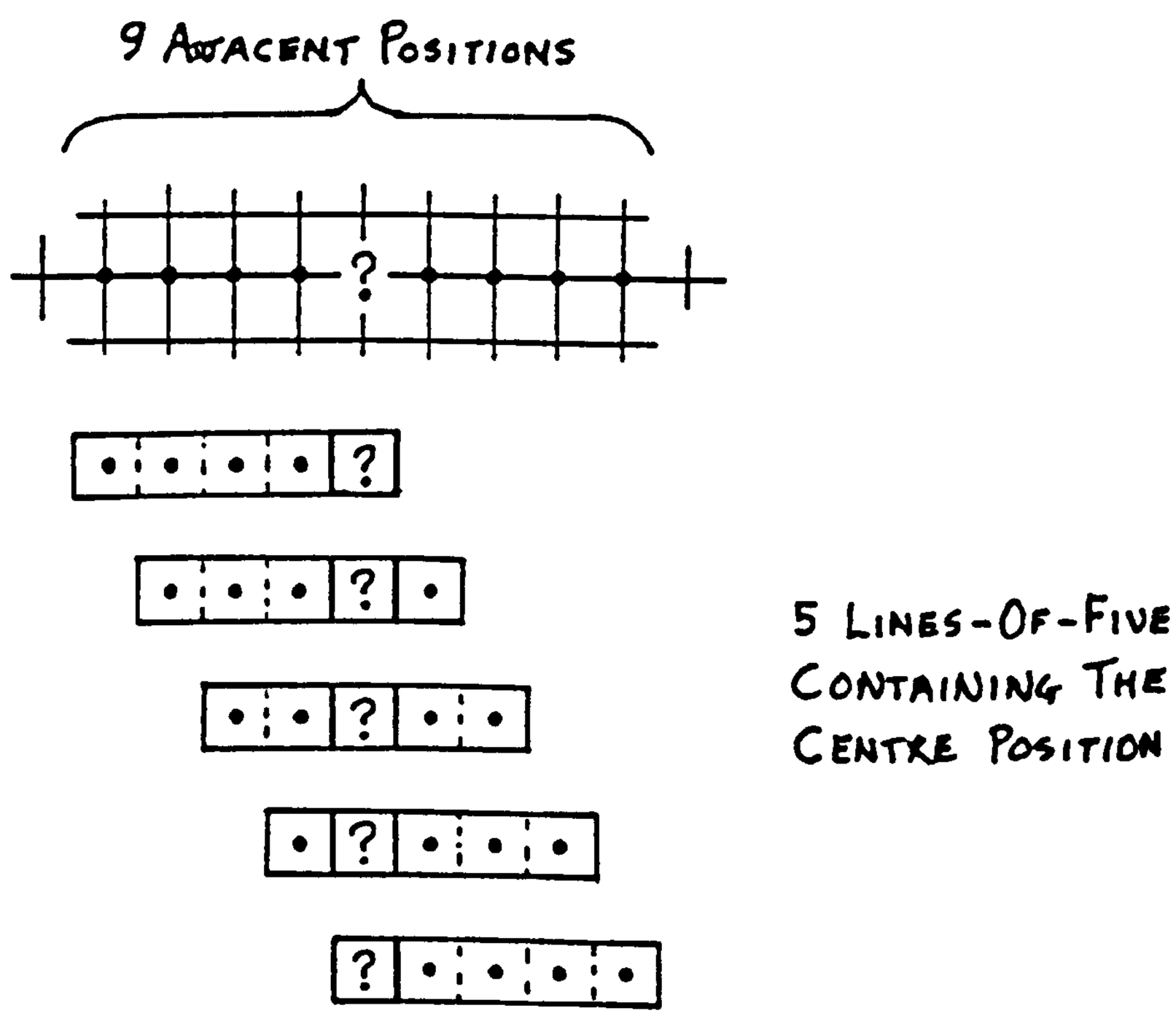


Figure 3.2 Adjacent Positions within Lines of Five

From Fig. 3.2 it can be seen that if '?' is the move position then:

PM (or OM) can have the values 0 (no occupied positions in the line of five), 1, 2, 3 or 4 (all other positions in the line of five occupied). Also PL (or OL) can have only the values 1 (a single line contains PM (or OM) stones), 2, 3, 4 or 5 (all five lines have PM = 0 (or OM = 0) stones, as is the case in Fig. 3.2).

Therefore, three bits are required for PM (and OM) and three bits for PL (and OL). Six bits are used for the PL.PM (or OL.OM) pairs. For each [/]d code 2 x (3+3) = 12 bits are used, thus for all four directions in the full code 4 x 12 = 48 bits are used.

Note that 48 bits is very redundant. Only the 71 [/]d combinations given in Appendix C can occur, so non-redundant coding of [/]d codes would require only 7 bits giving 4 x 7 = 28 bits for the full code. Carrying the coding efficiency further, only 71^4 full codes can occur, requiring only $\log_2(71^4)$ or 25 bits to specify each uniquely.
For coding simplicity 48-bit codes are used but note that with the learning net structures used (and described in Section 3.2.4.3) less than 2^{48} addressable RAM locations will be used (see Section 3.2.4.2). Note also that the redundancy introduced helps the net to generalise (Section 3.2.4.2).

It should be noted that not all board positions within a radius of five from the move position are included in the coding. Therefore any system based on this coding will be "blind" to all the positions marked with a cross in Fig. 3.3 when valuing the central move position. The most serious of these "blind spots" for Kenju are the "knight's move" positions marked with an 'x'.

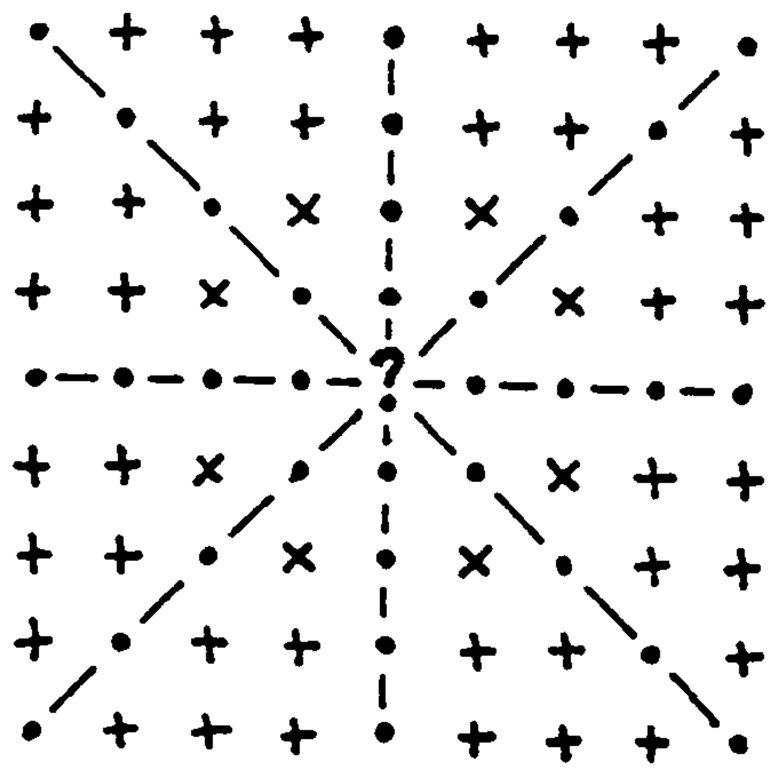


Figure 3.3 Blind-spots

3.2.4 The Learning Net

The learning net is an assembly of Random Access Memories (RAMs) used as a network of adaptive memory elements as explained in Aleksander, Stonham and Wilson (1974).

3.2.4.1 Single Element Nets

The basic block of 2^N locations has N address terminals, a single output terminal, and teach-data and teach-clock terminals. Each location stores one bit.

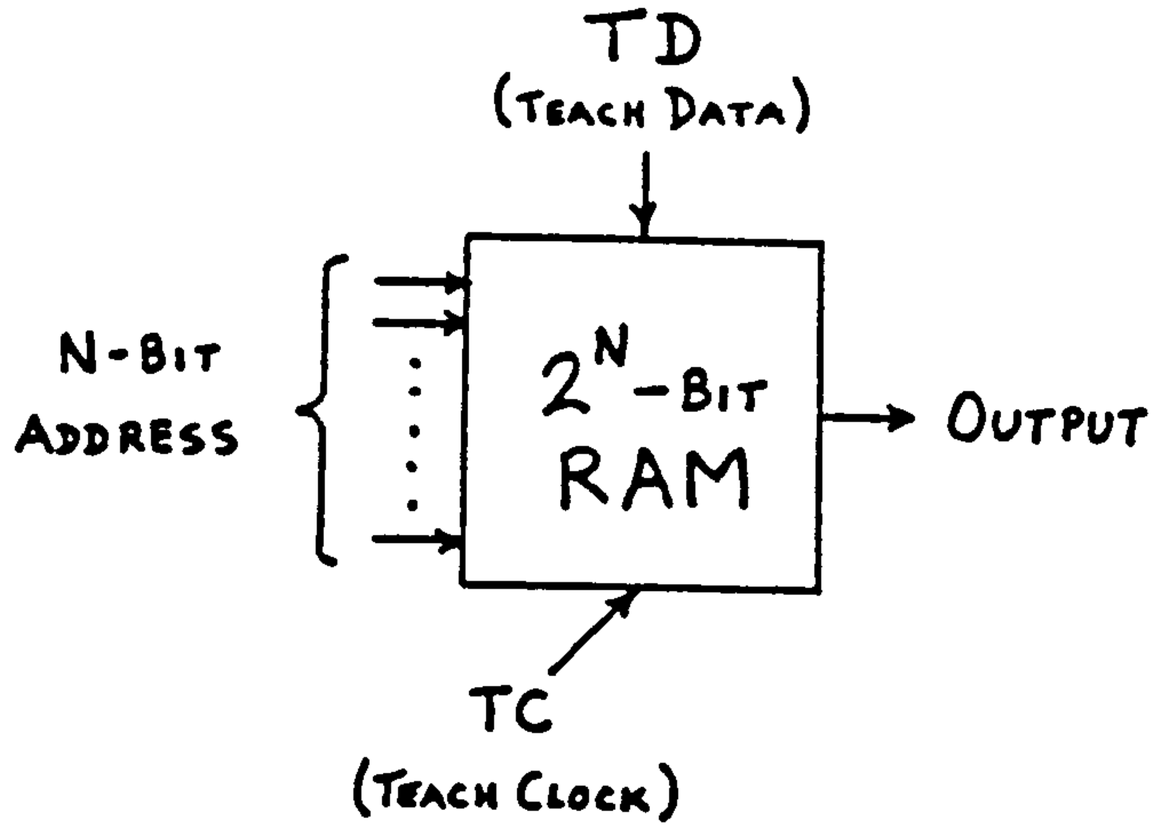


Figure 3.4 Single Element Net

The contents of the RAM locations are usually initially cleared to zero. Teaching then involves presenting to the address terminals an N-bit input pattern and forcing the desired output response by applying it to the TD (teach-data) input while clocking TC (teach-clock). This causes the data applied to TD to be stored in that memory location addressed by the bit pattern on the address terminals.

On subsequent non-teaching use the TD and TC terminals are not used, but the same N-bit pattern applied to the address terminals causes the value stored during the teaching phase to be asserted at the output as the response to the input address pattern.

The single bit RAM element just described can be extended to a word organised RAM element by addressing M memory blocks (single bit elements) in parallel and training for M-bit responses on the M output terminals by applying M-bit wide words to the M teach-data terminals during teaching.

The learning element described above is simply a look-up table. A simple net, or look-up table, such as this is used in Systems 3 and 4 to evaluate line patterns. See Section 3.4.3.1. Learning nets become more interesting when more than a single memory block is used, each addressed and taught separately as in Fig. 3.5.

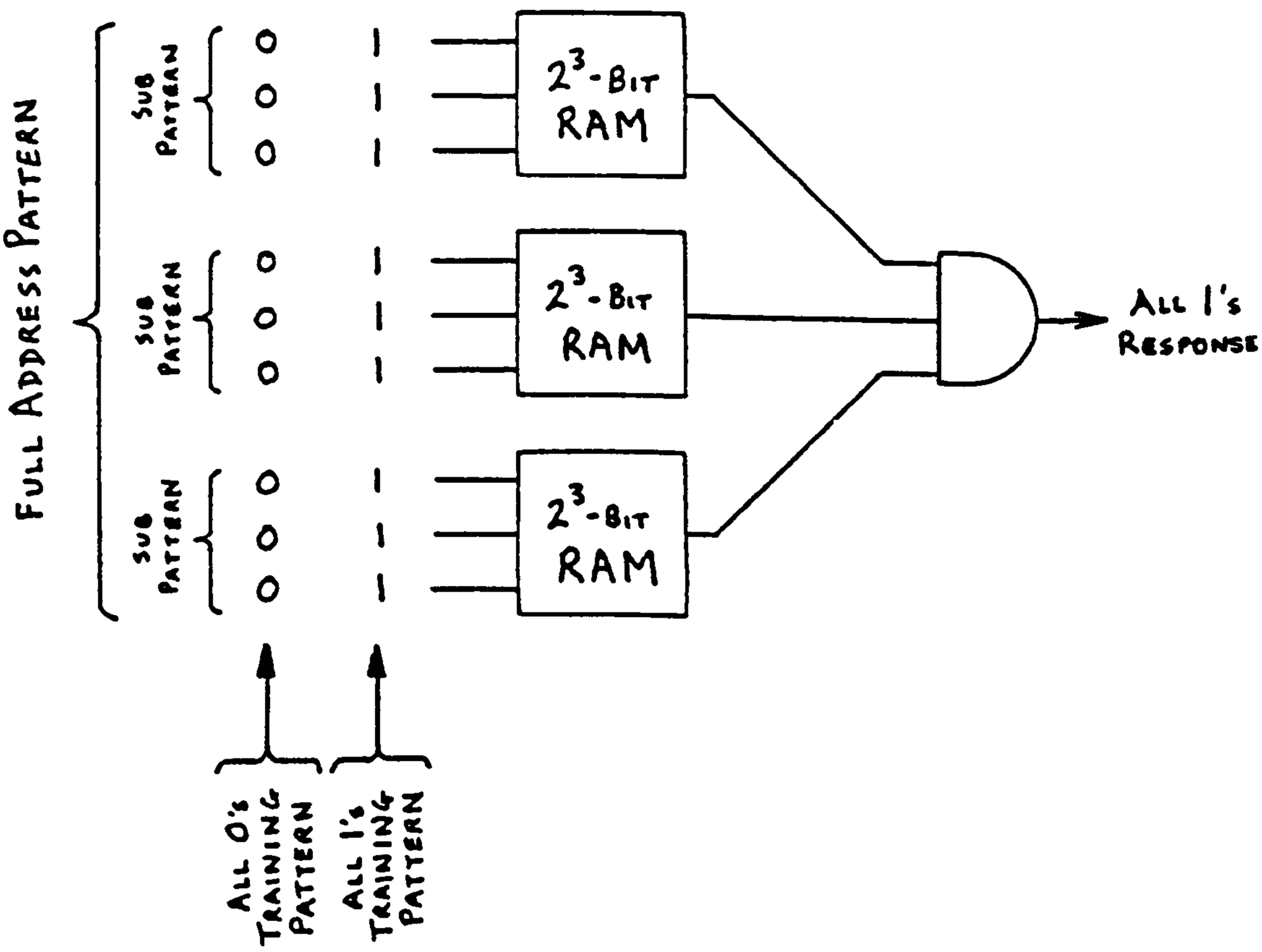


Figure 3.5 Multi-element Net

3.2.4.2 Multi-element Nets and Generalisation

The net in Fig. 3.5 effectively partitions the full 9-bit address pattern into three smaller, 3-bit, sub-patterns, resulting in an overall reduction in addressable memory locations ($3 \times 2^3 = 24$ locations whereas a single RAM element with a 9-bit address has $2^9 = 512$ locations).

Suppose the 3-way partitioned net of Fig. 3.5 is trained to give a response of all ones at its output for both the input patterns of all ones and of all zeros applied to the address terminals. The effect of having reduced the addressable locations is that application of patterns other than the all ones pattern or all zeros pattern at the address terminals will also give the trained response of all ones at the output terminals. In this particular example any cross product of the training sub-patterns will address, in all three RAMs, locations taught to respond with ones. Thus any of the full patterns below will give a response of all ones when applied to the address terminals.

000	000	000	— all zeros training pattern
000	000	111	other (non-trained) patterns
000	111	000	
000	111	111	
111	000	000	
111	000	111	
111	111	000	
111	111	111	— all ones training pattern

This effect of cross product patterns (not "seen" during training giving the same response as the training patterns constitutes "generalisation" by the net (Aleksander (1971), Aleksander, Stonham and Wilson (1974)). Generalisation can be beneficial as long as no two cross product patterns giving the same response are required to be distinguished.

A second type of generalisation can occur when several nets are used, each as a discriminator trained to recognise a particular class of patterns. This is the case with the net used in later versions of System 2 (described in Section 3.3.5). In this case generalisations can also occur if a particular discriminator does not respond with all ones (as for training patterns), but responds most strongly (i.e. with most ones) of all the discriminators. The pattern on the address terminals will not be a training pattern but is classed according to the most strongly responding discriminator. This discriminator would have one or more RAMs addressed by patterns not "seen" in a training pattern (and hence addressing locations not trained to output the response of one). The remainder of the RAMs in the discriminator would still respond with ones, giving the strongest overall response of any discriminator.

3.2.4.3 Net Used

The net structure for the initial system is one in which the situation codes for the four directions (arranged in order as described in Section 3.2.3 above) address four distinct 12-bit-address nets as shown in Fig. 3.6.

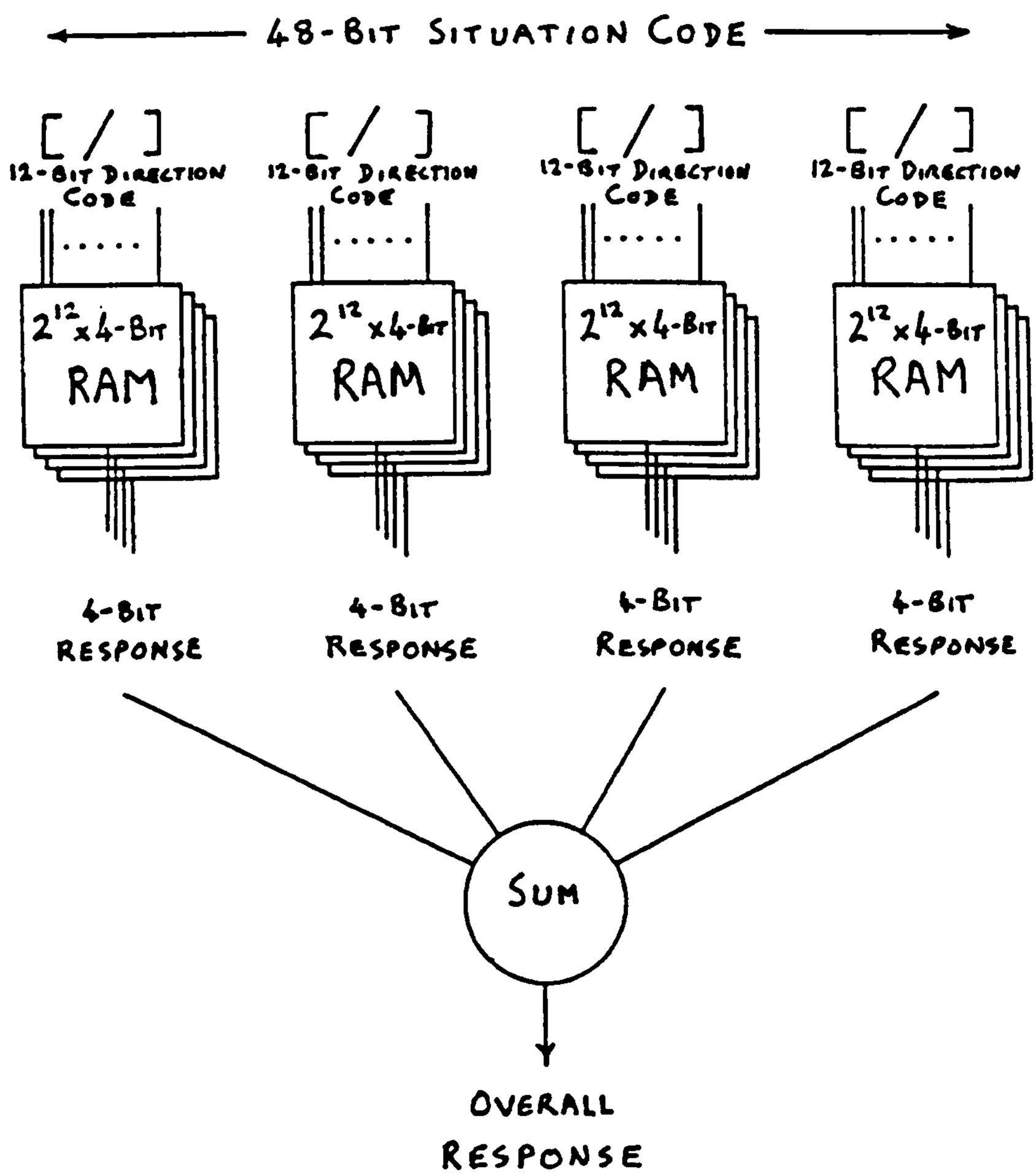


Figure 3.6 Net Used Initially

3.2.5 Move Response

Each 2^{12} - word net gives a 4-bit word response interpreted as a (positive binary number) weight, which is added to the responses for the [/]d sub-codes of the other three directions to give the overall response of the net to the full 48-bit situation code.

3.2.6 Best-Move Selector

The move position which gives the maximum response from the net out of all the moves in the useful-moves store is chosen for the computer player's move. When two useful-moves give the same maximum response the first move evaluated is taken, arbitrarily, as the computer player's move.

3.3 System 2

3.3.1 Introduction

The pilot system of the previous section was developed into a second system: System 2 represents the furthest state of development of the initial system model. The description which follows encompases many modifications, and hence several system versions. They are all discussed under the heading of System 2. But first a brief description of an underlying hypothesis for the second system is given.

3.3.2 Hypothesis

The underlying hypothesis for System 2 is outlined below. Although only part of the hypothesis has been implemented and tested it has given a direction to the development of the system.

A hypothesis may be loosely stated by means of the following statement and elaboration in the paragraphs which follow:

Intelligence in the model of a Kenju player is based on the interaction between the build-up of knowledge and the use of knowledge.

3.3.2.1 Built-In Abilities

The build-up and use of knowledge relies on certain built-in capabilities (which one can call "preprogrammed") as well as a learned ability.

Built-in abilities are preprogrammed to:

a) assemble all feasible moves (e.g. useful-moves selector of Sections 3.2.2 and 3.3.3),

b) extract vital parameters for each feasible move (e.g. situation coding at Sections 3.2.3 and 3.3.4), and

c) assess vital parameters using a learning net where logic is not built in.

3.3.2.2 Build-Up of Knowledge

Knowledge is acquired through learning by:

a) building up logic in the learning net, and
b) the setting of levels in decision mechanisms.

Learning is accomplished by specific teaching; all taught moves being strongly "rewarded" by a teacher. This implies that a mechanism is required whereby the teacher can indicate the essential move at a particular point of play.

3.3.2.3 Use of Knowledge

The built-up knowledge is used during the choice of moves and is assessed (and modified) in relation to its longer term success in improving the player's position in the game.

1) The top X candidates for a move (random if equal) are selected and ordered by assessing the parameters of each of the feasible moves.

2) Each of the top X candidates is re-evaluated over Y look-ahead moves, following only the "best" branch at each node (look-ahead move) in the tree of possible future moves. Re-evaluation assigns to the move the value of the best look-ahead move encountered during look-ahead.

3) If the ordering of the top X candidates is changed through re-evaluation the system should attempt to correct this by self-teaching.

4) Candidates are considered equal if they differ in value by less than a threshold Z. Thus Z difines when random choices are made in (1). Z is changed by the system itself depending on the success or failure of the current strategy, it thus implies a measure of uncertainty.

X and Y are independent (externally set) parameters while Z is a dependent (internally set) parameter.

3.3.2.4 Discussion

This hypothesis as a whole was never fully used to design a complete system, but was important in giving structure to the programming of System 2 (and parts of the systems which followed).

The parameter, Z, was found to be redundant since the range of move values was not large enough to require its use.

No basis has been found for implementing the self-teaching mechanism in (3) above. It would be sufficient to teach the best look-ahead value as the self taught value for the move which lead to it, except that this would tend to all moves which lead eventually to a win being valued the same as those moves leading to an immediate win. This point has been overcome successfully by Elcock and Murray (1967,1968) using the expected number of moves to a win, and by Koffman (1968) whose program learns to recognise (without valuing) moves leading to forcing sequences.

In System 2, teaching by indicating the essential move at a particular point of play (Section 3.3.2.2) has only been approximated to by the teacher weighting a good move when it is the move taken. See Section 4.4 for details. System 4 includes a closer approximation to teaching by indicating essential moves.

The change from System 2 to System 3 can be seen as a change of emphasis from the use of learned knowledge to more elaborate built-in capabilities which were considered necessary for further work. System 3 also represents the fusion of the built-in abilities, build-up of

knowledge and use of knowledge, into a composite search-and-learn process which learns as it searches for a move (Sections 3.4.4 and 4.5.1).

A description of the second system follows under the headings of the sections of the initial system which have been changed. Refer also to Fig. 3.1.

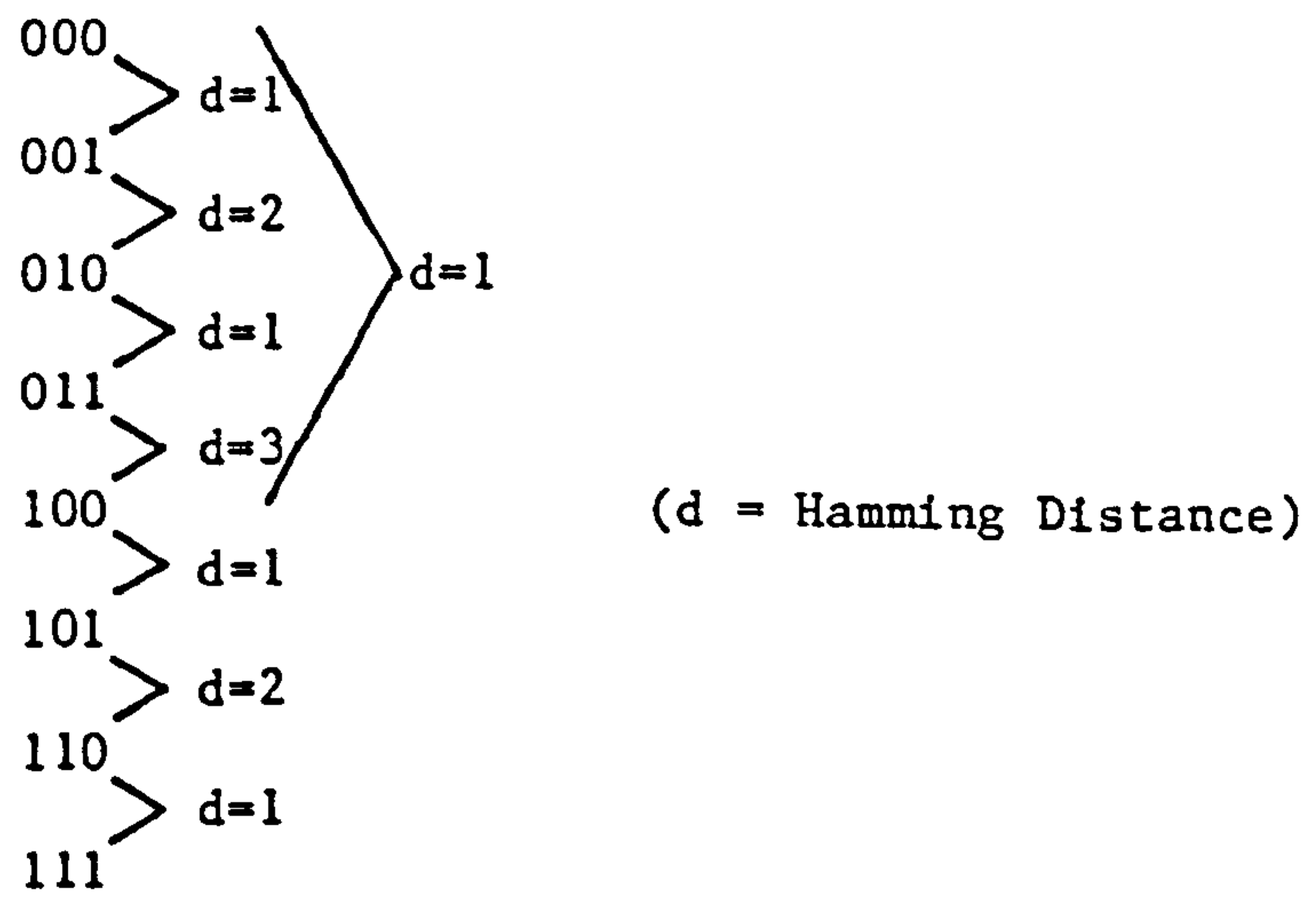
3.3.3 Useful-Moves Selection

Restricting the selection of places for the useful-moves store to only those in lines containing the most recent moves of the player and his opponent (i.e. the two previous moves) improves the continuity of the game. Each move tends to be either a continuation of the player's attack or a defense against his opponent's last move. This also improves the execution speeds of the algorithms (by reducing the number of moves in the useful-moves store) and seems more similar to the way the author chooses moves. These advantages offset the disadvantage of introducing "blindness" of the player to areas of the game which have not recently been active. Uhr (1973) uses the same mechanism in his GO program with direction to its moves.

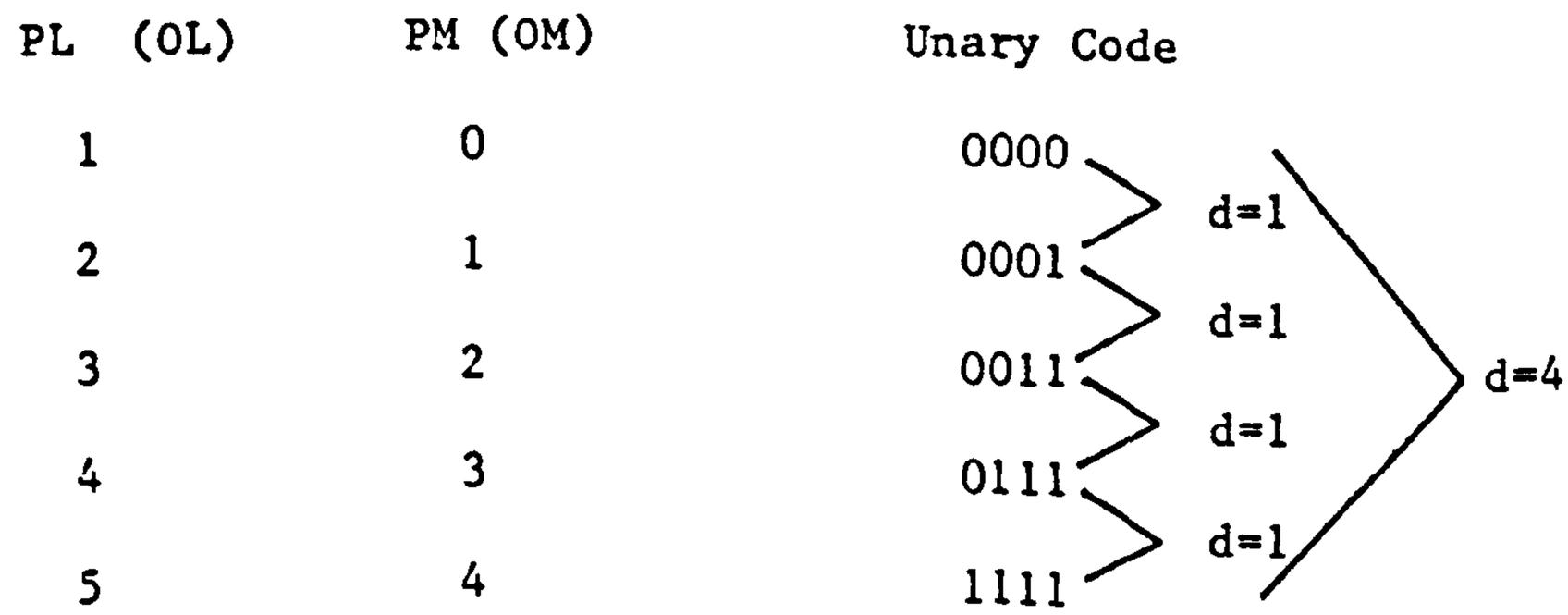
3.3.4 Moves Situation Codes

RAM learning nets do not respond well to patterns whose degree of similarity is not reflected in the Hamming distance between them. (Hamming distance is a measure of the number of bits which differ between two patterns.) This is because a (non-training) pattern differing, in a number of bit-positions, from a pattern "seen" during training will give the trained response on all those RAMs whose inputs are not connected to the differing bits in the pattern. Thus if the number of differing bits is low (the patterns are similar in Hamming distance terms) a corresponding low number of RAMs will be wrongly addressed, and hence the response of the net (or discriminator) will be high (most of the RAMs will respond as for the similar training pattern).

Binary coded numerical values (which arise naturally out of using a standard digital computer) do not have a Hamming distance similarity related to the similarity between the values of the codes.



Stonham (1974) uses Grey coding of numerical data, but this coding does not retain the numerical ordering which is used for "biasing" the situation code for the "pattern-background" attention mechanism of Section 4.4.1. In forming the situation codes a simple unary coding is used which does retain the normal numerical ordering (when interpreted as a binary number by the computer), and has the required Hamming distance similarity between codes.



To accommodate the 4-bit unary codes for PL, OL, PM, and OM, the full situation code must be 4 x (2 x (4 + 4)) = 64 bits.

3.3.5 Changes to the Learning Net

The initial word-organised net (System 1, Section 3.2.4.3) is inflexible in training due to the binary coded, numerical value, responses.

The modified structure, shown in Fig. 3.7 has eight independent discriminators (learning nets) replacing the single, word-organised, net. Each discriminator is allocated to the class corresponding to a particular response level. Thus, a particular discriminator can be trained to respond with ones on all RAM outputs when presented with those patterns which are classed as having the value associated with that discriminator. The number of RAMs of a particular discriminator which respond at their outputs with a one is thus an indication of the "similarity" of the pattern to the class of patterns to be valued at that level. (See the discussion on generalisation (the second type of generalisation) in Section 3.2.4.2). Hence the discriminator responding (with a one) on most RAM outputs gives the value to be assigned to the input pattern (move situation code), and so to the move being evaluated. The net of Fig. 3.7 hence has eight discriminators each consisting of eight RAMs with eight address inputs and one response output each. The address mapping selects, for each address input of a discriminator, one of the 64 pattern bits of the full move situation code. The selection is done on a randomised basis and all discriminators have the same address mapping. The particular form of this mapping is discussed in Section 4.4.1. Since the structure of all eight discriminators is identical, discriminators differ only in their trained responses (contents).

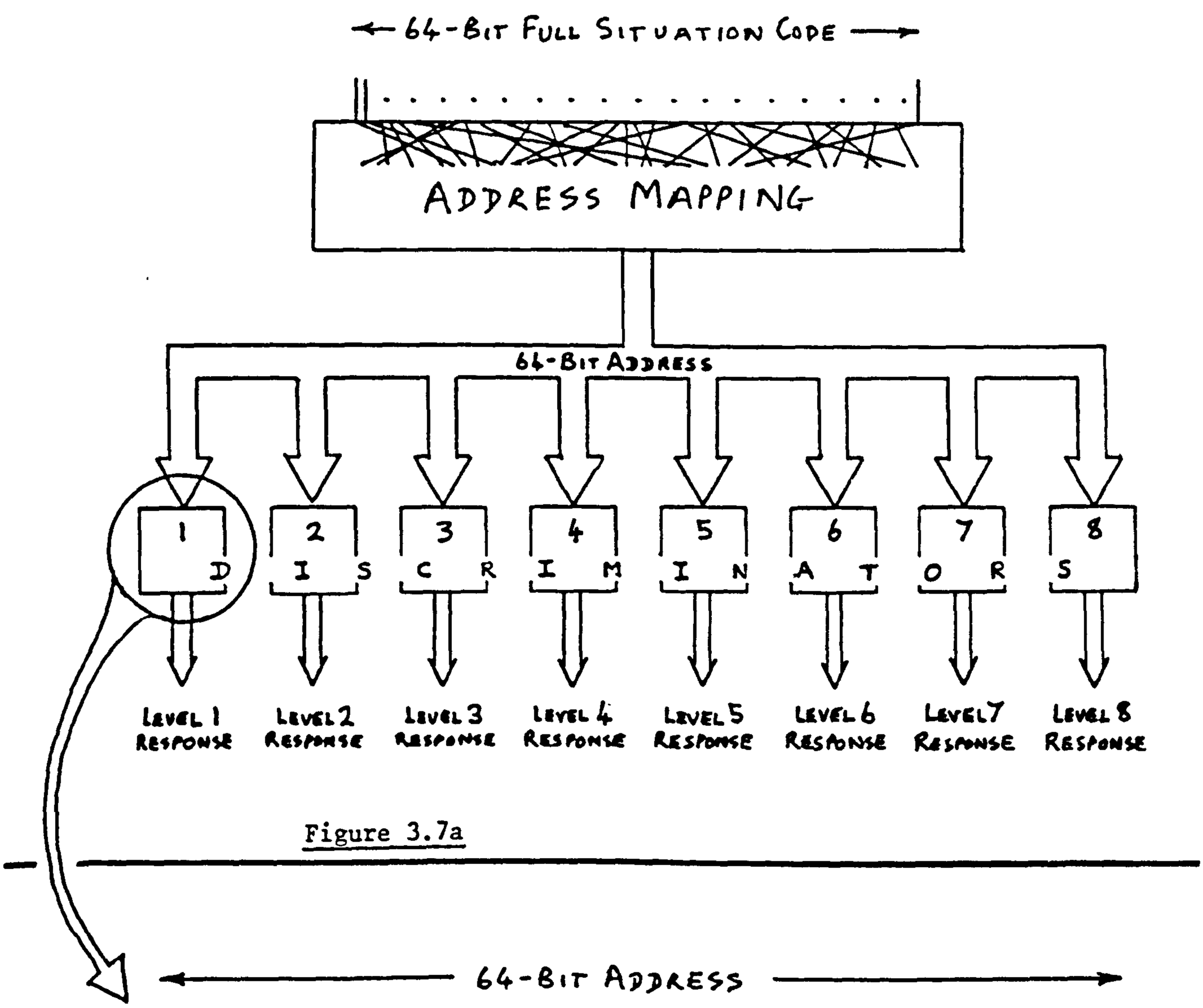


Figure 3.7a

Figure 3.7b Discriminator Nets

3.3.6 Move Response

The response to a move evaluated by the net of Fig. 3.7 is taken to be the value associated with the discriminator responding with most ones on its RAM outputs. In the case where a set of discriminators all give the same best response, the value assigned to the move is that of the highest valued discriminator of the set.

3.3.7 General Comments

Systems 1 and 2 use RAM learning nets since the task of move evaluation can be thought of as one of classifying the board situation surrounding a move in terms of its value in the game. Pattern recognition by RAM nets seems appropriate for this classification task.

The alternative approach whereby situations encountered during previous play are stored in coded form with an associated value (Elcock and Murray (1967, 1968), rote learning in Samuel (1959)) has an indefinite storage requirement which seems, to the author, less likely as a model of real (human) knowledge representation than does the fixed storage, learning net, model.

3.4 System 3

3.4.1 Introduction

The third system involves a complete change from the move selection algorithm of System 2. It is more closely related to the search of a human player (the author in particular) than the previous algorithms, but is a less obvious computer solution. Many of the problems arising out of previous algorithms for the move search are avoided by System 3.

The system will be discussed as a whole since it does not bear much resemblance to the systems explained above. The overall structure of the system is depicted in Fig. 3.8. The opponent move selection, and the move taking mechanism are the same as in the previous system (Chapter 2). The move search has completely changed and is explained here.

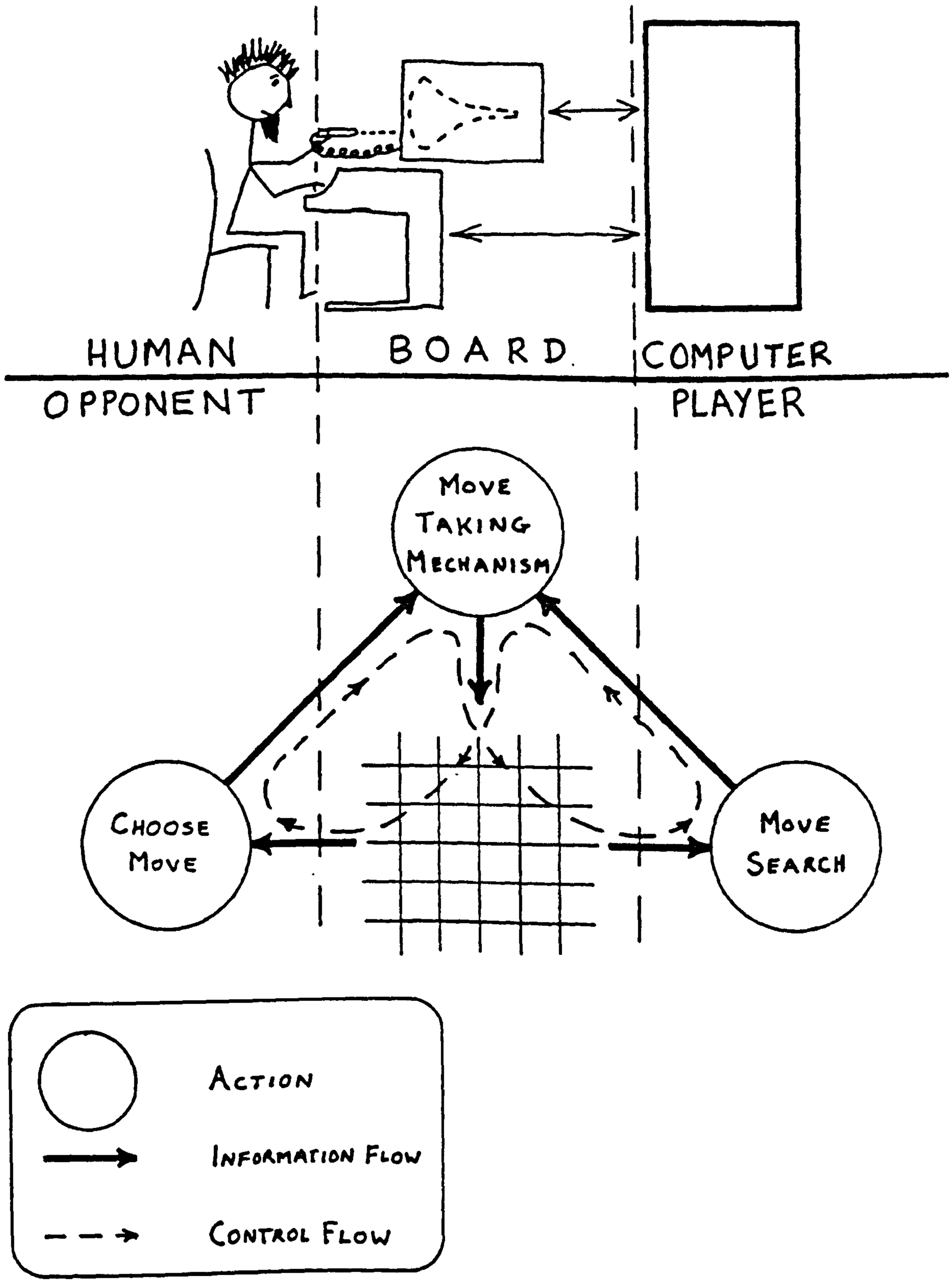


Figure 3.8 System 3 Block Diagram

3.4.2 Move Search

The computer player's move search models the exploratory path, or chain, of a player's attention across the board while looking for a best move.

"Seeing" a line leads to "attention focussing" on a move in that line; which, in turn leads to "attention focussing" on other lines intersecting (crossing) at that move. Attention then focusses on one of these intersecting lines; the process repeats with attention focussing next on a move in this line, and so on. Refer to Fig. 3.9 for an example. Also see Figs. 3.13 and 3.14.

At each step the "best" move or "best" line is chosen, where "best" has the significance explained below.

It should be noted that although this looks similar to mechanisms used for look-ahead (Uhr (1973)) it is simply a means of generating a chain of attention for the player, and therefore differs from look-ahead in that it does not involve moves of the opponent or predicted future play.

An example search path is given in Fig. 3.9.

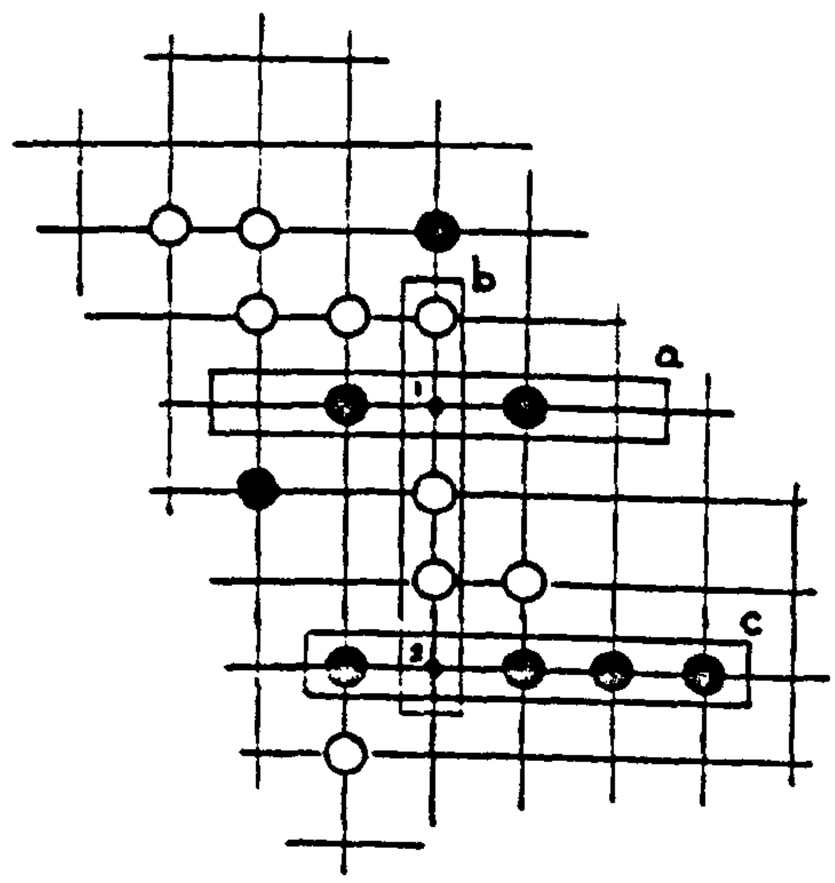


Figure 3.9 An Example Search Path

A possible line - move sequence shown in the search path of Fig. 3.9 is:

a -> 1 -> b -> 2 -> c ...

Assuming line a to be chosen at random, then move 1 is the best within line a. Line b is the best line intersecting at move 1, and move 2 is taken to be the best move in line b. Line c is then the best of the lines which intersect at move 2.

3.4.3 Line Evaluation and Move Evaluation

The move search of System 3 is based on the following line and move evaluation methods, which are required for boxes 2 and 6 in the move search flow chart of Fig. 3.12 and will therefore be described here before the detail of the flow chart (in Section 3.4.4).

3.4.3.1 Line Evaluation

Line patterns are evaluated by a simple learning net using a single word-organised RAM with five address inputs (see Section 3.2.4.1) as shown in Fig. 3.10.

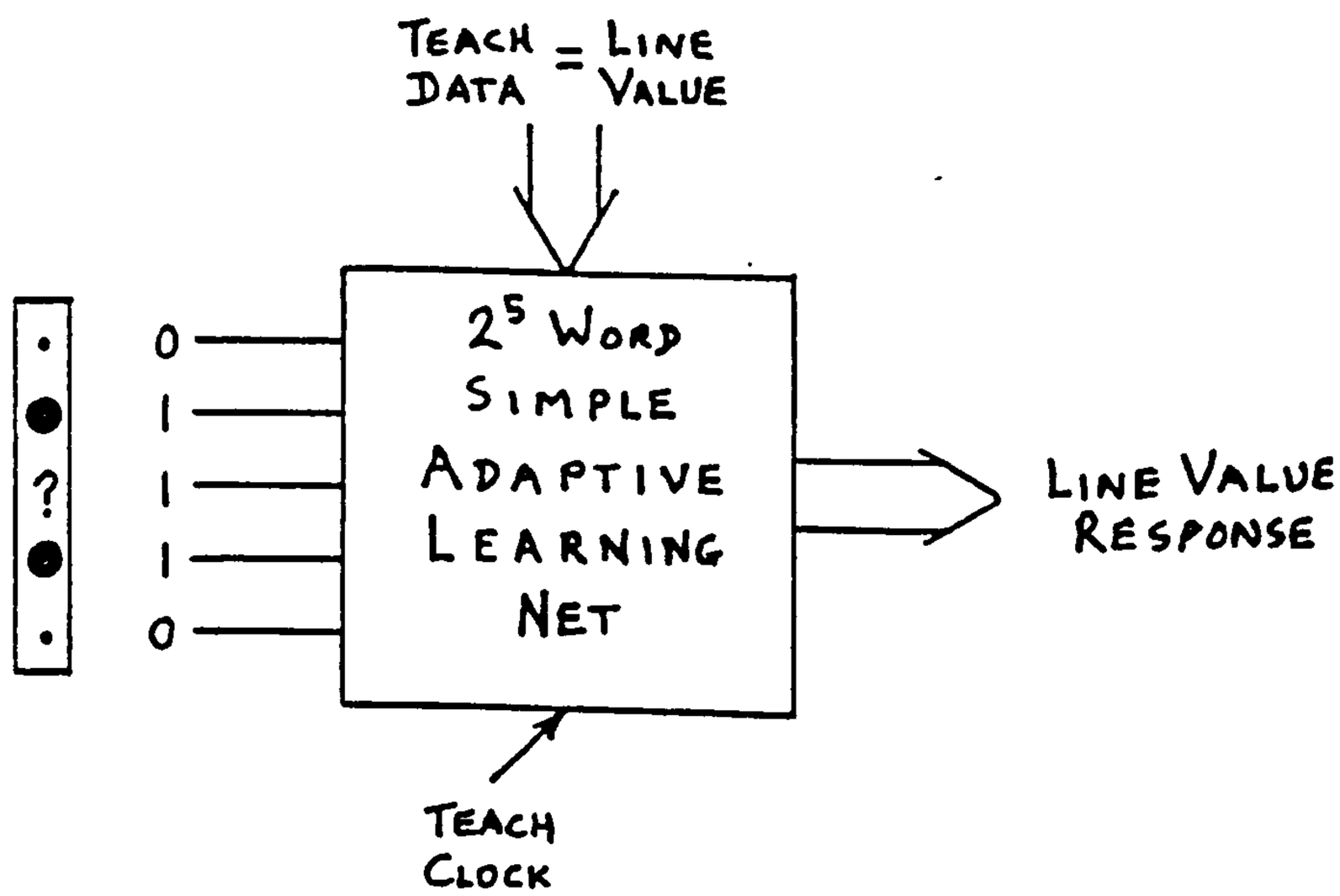


Figure 3.10 Line Evaluation Net

The 32-word RAM of Fig. 3.10 acts as a look-up table with the pattern of vacant and occupied positions in the line forming the table index. The taught response on the output is a binary coded numerical value. In the system implemented, occupied positions (stones) in the line are signified by ones and vacant positions by zeros. The "?" signifying the position of the player's possible next move is considered as an occupied position in that it is coded as a one. The net thus evaluates the line as it would appear after the move.

3.4.3.2 Move Evaluation

Moves are evaluated in terms of the values of lines which intersect at the move. That is, the value of a move is some combination of the values of those lines to which the move contributes. The example of Fig. 3.11 is used to explain the evaluation of the move position signified by "?".

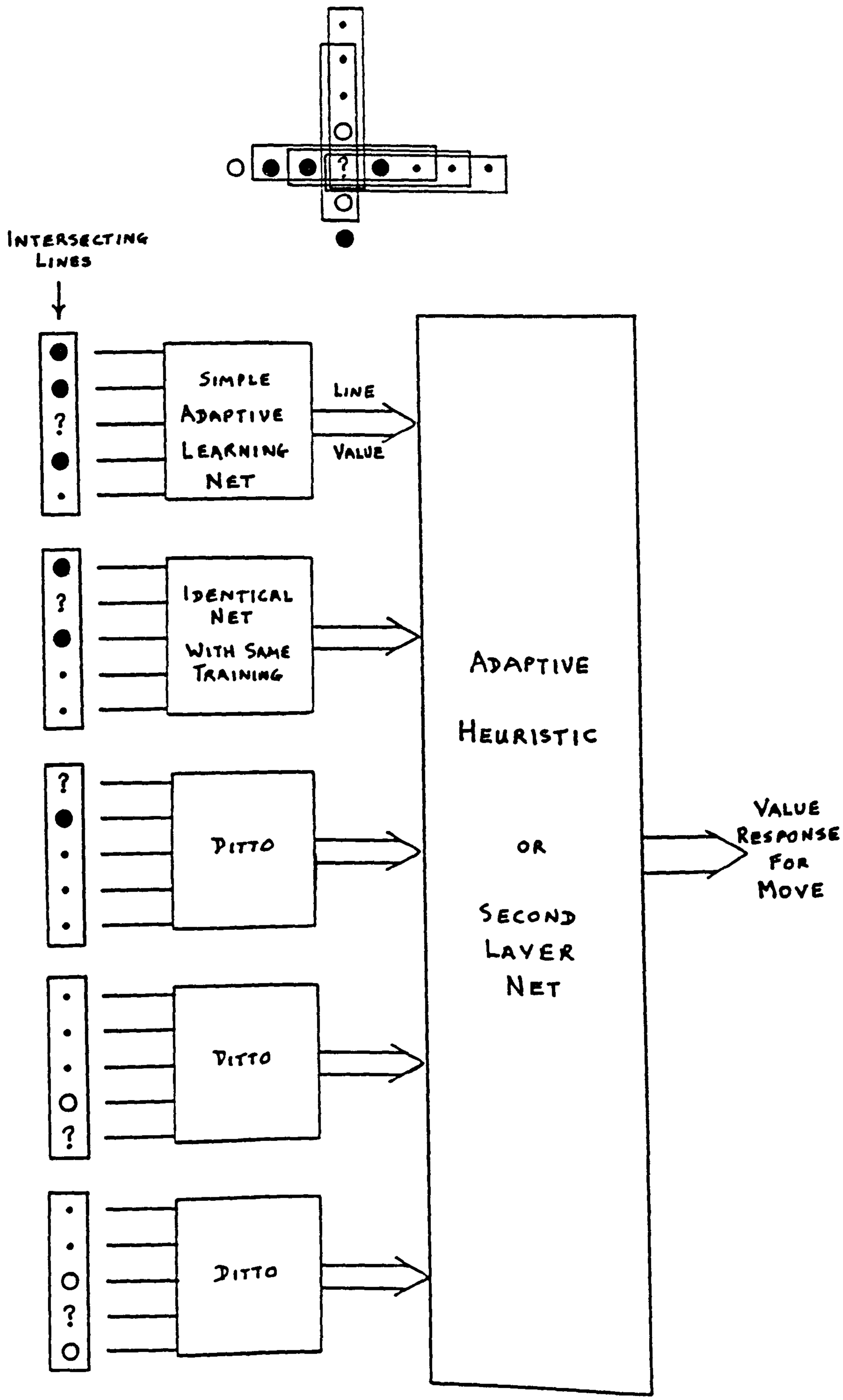


Figure 3.11 Move Evaluation Example

To evaluate a move, values are obtained (by line evaluation, Section 3.4.3.1) for each of the lines intersecting at the move position (a possible maximum of 20 lines). These line values are then used to form a value for the move. This latter stage of evaluation uses a (potentially adaptive) heuristic in the existing implementation, but might be possible using a second layer learning net.

3.4.3.3 Move Evaluation Heuristic

The heuristic can be stated as follows:

<u>if</u> any one line intersecting at the move has a sufficiently "<u>outstanding</u>" value (determined by a threshold)

<u>then</u> its value is taken as characterising the value of the move (e.g. in the case of an almost complete winning line)

<u>else</u> take the value of the move to be the sum of the values of those intersecting lines which have values above a "<u>mediocre</u>" threshold.

The setting of the two thresholds in the heuristic is determined by the teacher in the existing system but could possibly be done dynamically by the learning algorithm (as will be discussed in Section 4.8.4).

If a second layer of learning net is used to replace the heuristic it must be trained to reflect a similar heuristic. For example the above heuristic could be replaced by the method used in Elcock and Murray (1967) where the best, or best and second best, lines are included in the evaluation of a move.

Such a second layer net would be required to distinguish between patterns formed from the outputs of up to twenty first layer nets. This first layer would be twenty identically trained line evaluation nets similar to those of Section 3.4.3.1 but with Hamming distance coded response outputs (see Section 3.3.4). The second layer net would have to contend with patterns varying in (significant) length according to the number of intersecting lines. Neither this problem, nor the problem of making the net sensitive to the case of a single "outstanding" line (while retaining properties of generalisation) have been tackled. Initial investigations indicate that a learning net is probably not the right tool for the job.

3.4.4 The Move Search Flow Chart

The search flow chart of Fig. 3.12 can be summarised by the following:

Box 1:	Find, on the board, an area of interest (i.e. containing a good line).
Boxes 2+3+4+5+6:	Form a single search path through the search tree until a "satisfactory" move is found, or a dead-end situation is encountered.
Boxes 5+7+1:	A dead-end path results in a restart of the search with a new path.
Box 8:	Take the "satisfactory" move, or the best move encountered during a maximum number of dead-end paths.

The flow chart is explained box by box in the sections which now follow and should be read in conjunction with Fig. 3.12.

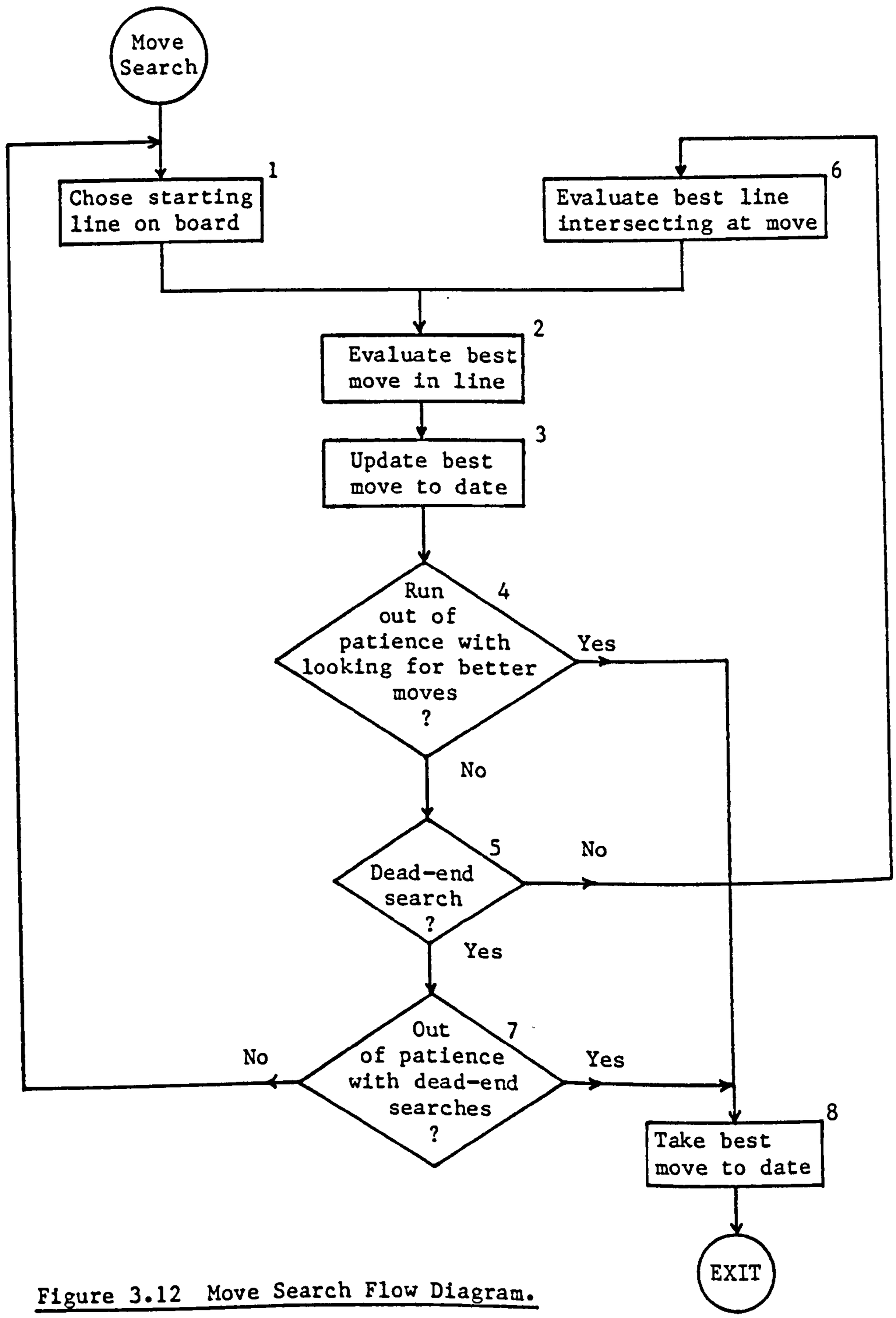


Figure 3.12 Move Search Flow Diagram.

3.4.4.1 Choice of Starting Line (Box 1)

A random selection is made from a subset of the lines on the board in order to choose a starting line for a search path. The lines included in this subset are controlled by the two parameters, DEFENDING and ATTACKING, explained below and in further detail in Section 4.8.1.

If the player is in "defensive mode", signified by the DEFENDING parameter, then the subset of lines from which a starting line is randomly selected will include those lines intersecting with his opponent's previous move.

If the player is in "attacking mode", signified by the ATTACKING parameter, then the subset of lines from which a starting line is randomly selected will include those lines intersecting with the player's previous move.

If both DEFENDING and ATTACKING parameters are switched on then both defending and attacking lines are included in the above subset. If the player is in neither DEFENDING nor ATTACKING mode a random selection is made from all lines on the board.

The purpose of the DEFENDING and ATTACKING parameters is to direct the search to specific lines, and hence moves, around the area of most recent activity. Section 4.8.3 explains how these parameters are set.

The advantage of this scheme over that employed in System 2 (Section 3.3.3) and in Uhr's (1973) GO program with direction to its moves, is the independent control of defensive and attacking strategies which need

not be mutually exclusive in the opinion of the author.

3.4.4.2 Best Move Evaluation (Box 2)

The "best" move (vacant position) in a line is found by comparing the value of each move in turn with the current "best" for the line, and replacing the current "best" move whenever a "better" move is found (one giving a higher value under move evaluation). The move involved in the previous itteration of the search loop (i.e. Boxes 2+3+4+5+6) is excluded from this evaluation.

3.4.4.3 Updating the best-move-to-date (Box 3)

Each time a "best" move is found in a line it is compared with the "best" move encountered to date during this move search. The new move replaces the existing best-move-to-date if it has a higher value.

3.4.4.4 The Iteration-patience Parameter (Box 4)

In order to limit the number of iterations of the search loop (Boxes 2+3+4+5+6) the algorithm terminates when a maximum number of "unsuccessful tries" have occurred since the last "successful try" and the algorithm is said to have "run out of patience". In this context a "successful try" is an iteration of the search loop which "betters" the best-move-to-date, all other iterations being "unsuccessful".

3.4.4.5 Dead-end Searches (Box 5)

Several looping conditions cause a break out of the search loop (Boxes 2+3+4+5+6) and constitute a "dead-end" search:

A place-loop dead-end occurs when, during evaluation of the best move in a line (Box 2), the only vacant position in the line is the move considered during the previous itteration of the search loop. This move is the one which led to the current line, so is excluded from the choice of best move in the current line. This avoids the possibility of the path repeatedly looping back to the current node of attention and forming a single-node cycle in the search path.

A node-loop dead-end occurs when the search path loops back to the current best-move-to-date, thus encountering a previously encountered node in the search path. This would result in the search path endlessly looping back to this node, forming a multi-node cycle in the search path.

A line-loop dead-end occurs when, during best line evaluation (Box 6), the only line intersecting with the current move is the line which led to the current move in the previous itteration of the search loop. This line is excluded from the choice of best intersecting line for a similar reason to that given for place loops above.

If none of these dead-end conditions hold at the test for a dead-end search (Box 5), then the search loop (and hence the search) continues with best line evaluation (Box 6).

If any of the dead-end conditions hold at the test for a dead-end (Box 5), the current search path is abandoned, only the current best-move-to-date being retained. A check is then made (Box 7) to limit the number of dead-end searches (dead-end paths) to a maximum, set by the teacher in the present system.

3.4.4.6 Best Line Evaluation (Box 6)

The best line intersecting with a move is determined by comparing the value of each intersecting line with the current best line intersecting at the move. The lines are chosen for comparison in a random order. If a line is found to be "better" than the current best intersecting line (if it is valued higher by line evaluation) then it becomes the current best intersecting line. The line from the previous itteration of the search loop (Boxes 2+3+4+5+6), the "parent" node in the search path, is excluded from this best line evaluation.

Only those lines appropriate to the current setting of the DEFENDING and ATTACKING parameters (see Sections 4.8.1 and 4.8.3) are considered (for best line) in best line evaluations.

That is:

<u>if</u> DEFENDING <u>then</u> lines of the player's opponent are considered

<u>if</u> ATTACKING <u>then</u> lines of the player are considered

<u>if</u> DEFENDING <u>and</u> ATTACKING <u>then</u> both the player's and opponent's lines are considered

<u>if</u> <u>not</u> DEFENDING <u>and</u> <u>not</u> ATTACKING <u>then</u> all lines are considered

3.4.4.7 Dead-end Patience (Box 7)

When a dead-end occurs in a search path (see Section 3.4.4.5) a check is made to ensure that a maximum number of dead-end paths have not already been followed before starting a new search path. This maximum is set by the teacher in the existing system, but could possibly be set according to the player's "interest in the game at this stage" or available time if a time limit is imposed. These possibilities are discussed in Section 4.8.5.

3.4.5 The Structure of the Search Path

The search path is a single path through a tree of "attention nodes" (the search tree) emanating from the initial starting line of the search path.

The search loop (Boxes 2+3+4+5+6) takes a line and evaluates the best vacant position (move) in it (excluding the "parent" position). The search path is then extended by choosing as the new starting line for the search process the best line (excluding the "parent" line) which intersects with this best move, as the new starting line for the search process. Fig. 3.13 represents a single node of the search path, formed by a single iteration of the search loop.

The two-stage best move + best line evaluation (the search loop) is repeatedly applied until the best move has been unchanged by the last N applications or a dead-end is encountered. N is regarded as a "patience" parameter. A representation of a search path is given in Fig. 3.14.

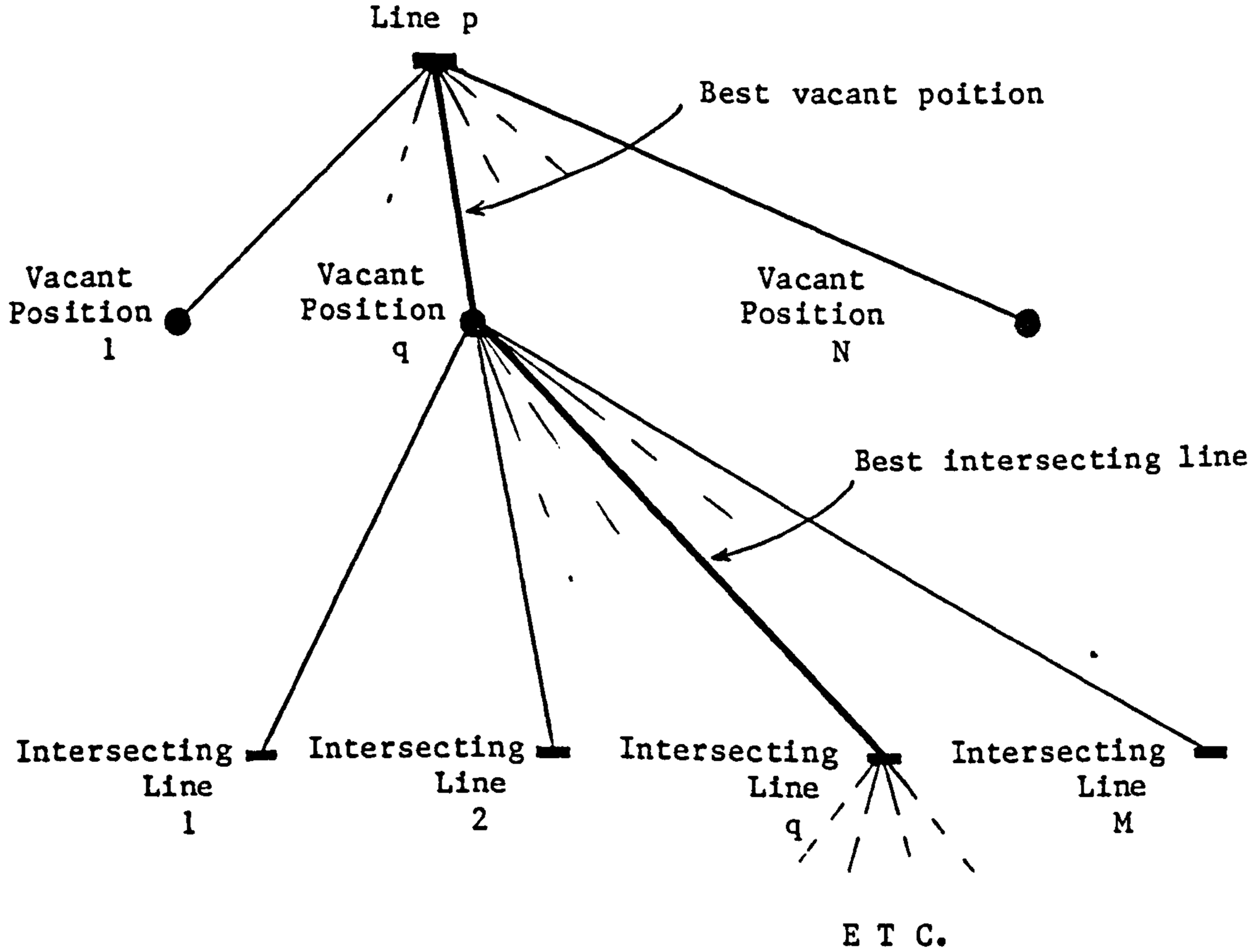


Figure 3.13 A Node of the Search Path.

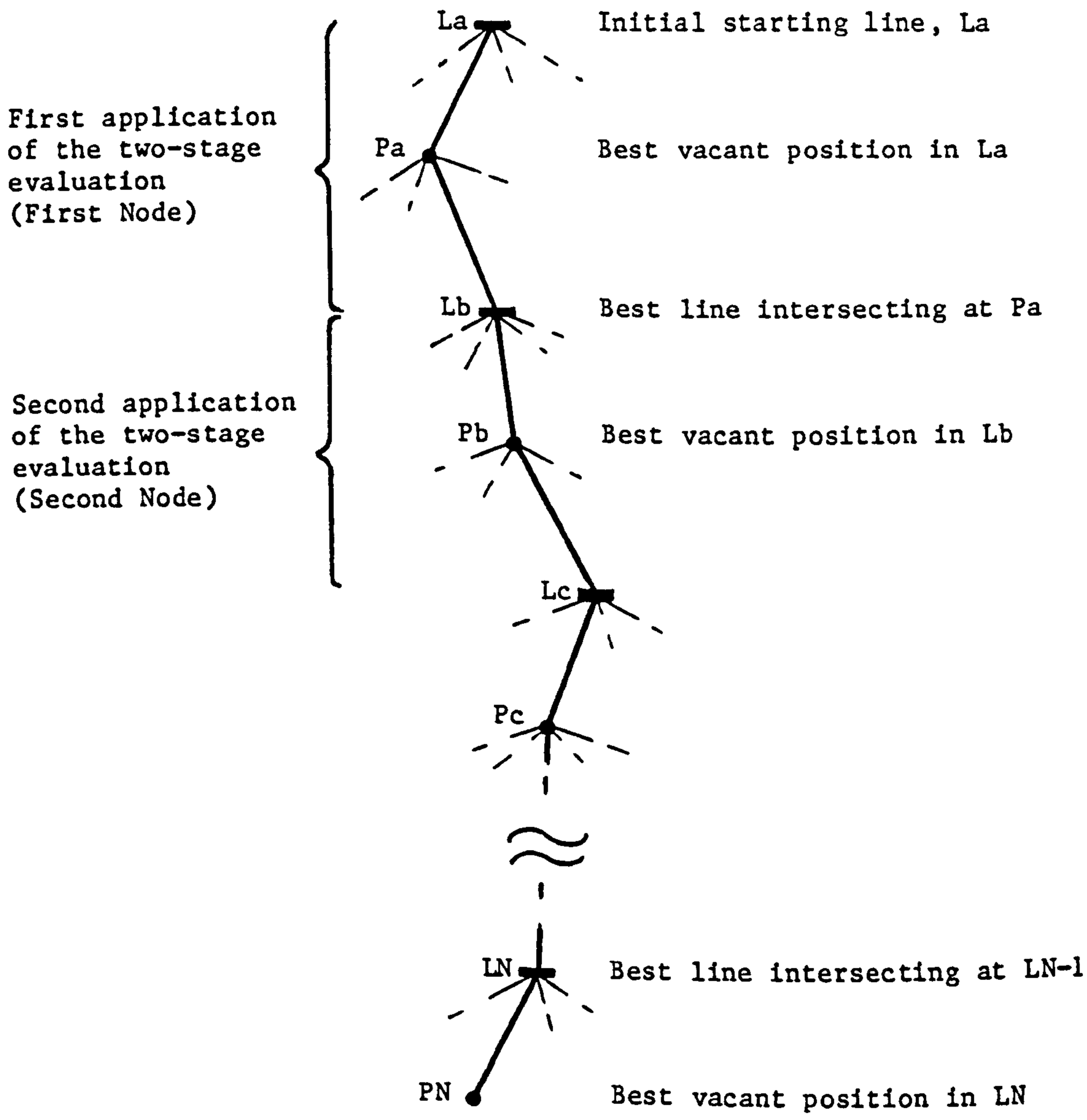


Figure 3.14 A Search Path.

3.5 System 4

3.5.1 Relation to System 3

Before reading what follows, the reader may wish to read, in addition to the section on System 3 in this chapter, the corresponding sections (4.5 and 4.6) of chapter 4, in order to fully understand the difference between System 3 and System 4, which is solely a matter of method of learning.

System 4 involves one small but significant change to the third system. This change is related to the way advice is given by the teacher and interpreted by the learning system. System 4 is in other respects identical to System 3.

In System 3 advice is requested of the teacher when the question of which is the better of two lines cannot be decided by the learning system on the basis of its present knowledge. See Section 4.6. Learning in System 4 has the same overall structure of requesting advice where an arbitrary decision would otherwise have to be made, but the teacher in System 4 is asked to indicate the better of two moves which the system has valued equally, rather than the better of two lines as in System 3.

This makes the teacher-learner interface more natural for the teacher and allows many learning actions to follow as a result of each piece of advice. How this new method of move oriented teaching fits into the line oriented learning is a matter for discussion in chapter 4 (section 4.10), but the essential philosophy behind it is explained here.

3.5.2 Teaching by Preferred Moves

The learning of System 3 is very explicit and direct in that the teacher essentially "tells" the learning net exactly what changes are to be made to the system's stored knowledge. This is both unrealistic in terms of what a human teacher tells a human learner, and in terms of the questions a human learner asks of his teacher and how he then responds to his teacher's advice.

Questions asked by a human player learning to play Kenju are usually in the form of questions about moves, although the teacher's answer may still be, in essence, a clarification of the learner's knowledge about useful lines. The specific information about which lines are essential in the teacher's preferred move is implicit in the teacher's advice but is not made explicit by the teacher. The learner derives for himself, by making assumptions about the advice given, the necessary modifications to his knowledge which are implied.

The learning which results from this advice represents the learner's assumptions about the advice. In this model such advice is interpreted by the learning system as indicating a set of preferred lines (line patterns) which intersect with the preferred move but are not present among the lines intersecting with the other moves. The learning derived from each piece of advice about moves thus relates to certain lines which intersect at the preferred move.

The assumption that the preferred lines are those involved in the preferred move but not involved in the other moves, is simply an assumption that the teacher is selecting the preferred move on the basis of preferred lines which are particular to it.

3.5.3 Summary and Discussion

To summarise: while learning is concerned with line patterns and their relative values, teaching on the other hand, is a matter of preferred moves.

With reference to the hypothesis of System 2 (see Section 3.3.2.4) System 4 could support "teaching by indicating the essential move at a particular point of play". That is, the teacher could take complete control of teaching by indicating (without being asked by the learner) which of a set of moves is the esential and preferred move at a particular state of a game. The larger the set of moves from which a preferred move is selected, the fewer would be the lines unique to that preferred member of the set and so the more specific the teaching resulting from such advice.

These two systems of learner-initiated and teacher-initiated learning could complement one another in a single system, possibly together with the self-teaching of section 3.3.2.3.

CHAPTER 4

LEARNING METHODS AND TRAINING

4.1 Introduction

Chapter 4 is intended to be read with reference to Chapter 3. Chapter 3 covers the structure of each of four learning systems whereas Chapter 4 describes the training procedures used (from the point of view of the teacher) and the internal workings of the learning elements (from the learner's point of view). Chapter 4 also presents experimental results, where they exist, and indicates the deficiencies of a method, where appropriate, with suggestions of changes required.

The major part of Chapter 4 relates to Systems 2 and 3. Experience gained from System 1 (the initial system) is incorporated in System 2. System 4 is in a similar way a modification of System 3. The results presented are for System 3. System 4 is an unimplemented proposal.

Each section of Chapter 4 indicates the corresponding sections of Chapter 3 where the system under discussion is described. In order to avoid too many references to Chapter 3 during the text, the reader is advised to familiarise himself with relevant sections of Chapter 3 before reading the corresponding section of Chapter 4. These relevant sections are indicated at the start of each section of Chapter 4.

4.2 Initial Investigations of Learning (System 1)

No results are presented for learning in System 1, but the methods used, and certain defects of them, are discussed. System 1 represents the initial pilot system which illuminated several deficiencies that had to be overcome. Section 3.2 describes System 1.

4.2.1 Training

The following training corresponds to the learning net of section 3.2.4.3, fig. 3.6 (see also section 3.2.4.1 for terminology used) and the system diagram of fig. 3.1.

The initial training strategy for System 1 is very crude, all the teach clocks of the learning net being clocked for every move of the opponent. This effectively causes the computer player to assume the opponent always makes a good move. The taught response of the four RAMs was simply the old response for each RAM (viewed as a 4-bit binary numerical value) incremented by one.

The effect of the above training scheme is to encourage the use of those situations which occur most often around the opponent's moves.

4.2.2 Deficiencies

A disadvantage of this training scheme arises since the commonest situations around moves in a normal game tend to be those of many empty, or near empty, lines surrounding the move. Hence, the computer player is observed to "leap" about the board in pursuit of unoccupied areas. This is clearly not a good strategy for winning Kenju.

The interaction of two properties of the net of System 1 account for this unsatisfactory behaviour: the inappropriate nature of its generalisations, in the context of what will be termed the "pattern-background" problem.

First, the generalising property of the net means that, even when an almost complete winning line of five exists in one direction, the full situation code tends to be dominated by the other [/]d sub-codes of the other directions. These other directions are often sparsely occupied and so not of great value.

Locations of those RAMs connected to the sparsely occupied directions are also (wrongly) trained to respond with a high value when a full code containing a useful [/]d pattern in one direction is rewarded. Thus, when the net subsequently evaluates a move completely surrounded by unoccupied (i.e. worthless) lines in all directions, the response is unintentionally high. The intention was to "reward" only the useful part of the full pattern, but useful patterns in one direction are almost always accompanied by non-useful patterns in other directions. The fault lies in training non-useful patterns along with useful patterns, but exhibits itself as a consequence of the generalising property of the net. Thus, the system misinterprets the reward as applying to the generalisation of rewarded patterns. A more specific method for rewarding is needed.

4.2.3 The "Pattern-Background" Problem

The second part of the problem with the net of System 1 is more fundamental. The net response is equally determined by bits in the situation code reflecting unoccupied places (background) as it is by those bits corresponding to occupied places (pattern). An analogy can be made with a similar problem in scene analysis where it is often difficult to say what is pattern and what is background. The net must be "biased" such that the response is determined more strongly by bits in the situation code corresponding to occupied places.

Since the four [/]d direction codes are ordered according to numerical value (section 3.2.3), then, by arranging that "heavily occupied" directions are represented by codes with higher numerical values than codes for sparsely occupied directions, one end of the full situation code will tend to reflect occupied patterns while the other, unoccupied background. This grading (by position) of direction codes according to degree of line occupation is broadly satisfied by the coding methods used (sections 3.2.3 and 3.3.4) except that the [/]d sub-codes are split into high order bits (i.e. PL.PM) and low order bits (i.e. OL.OM) for player and opponent respectively.

With the bits (sub-codes) corresponding to occupied (and hence useful) patterns "gravitating" to one end of the full situation code, the net can be configured to respond with more account being taken of the "occupied" end than of the "unoccupied" end. Two methods of "biasing" the net such that its response is determined mainly by this "occupied" end are considered now.

The first method achieves biasing by attaching most net address lines to the "occupied" end (i.e. by address mapping), the second by differentially weighting the individual RAM outputs.

The address mapping solution (see fig. 4.1) involves sampling more heavily from the "occupied" end of the situation code than from the "unoccupied" end when selecting address inputs for the net as a whole. Such a technique is used in System 2 but has not been fully evaluated (see section 4.4.1).

The output weighting solution to the biasing problem can be accomplished by differentially weighting the four RAM outputs before summation, and sampling addresses for each RAM from a single [/]d direction sub-code (i.e. as in fig. 4.2). This allows those RAMs sampling the "occupied" end to be given more weight in the summation and hence in the overall evaluation. Biasing by weighting the outputs of the RAMs has not been investigated.

Figs. 4.1 and 4.2 show diagramatically the address mapping and output weighting solutions to biasing respectively. They should be compared to the initial configuration of fig. 3.6.

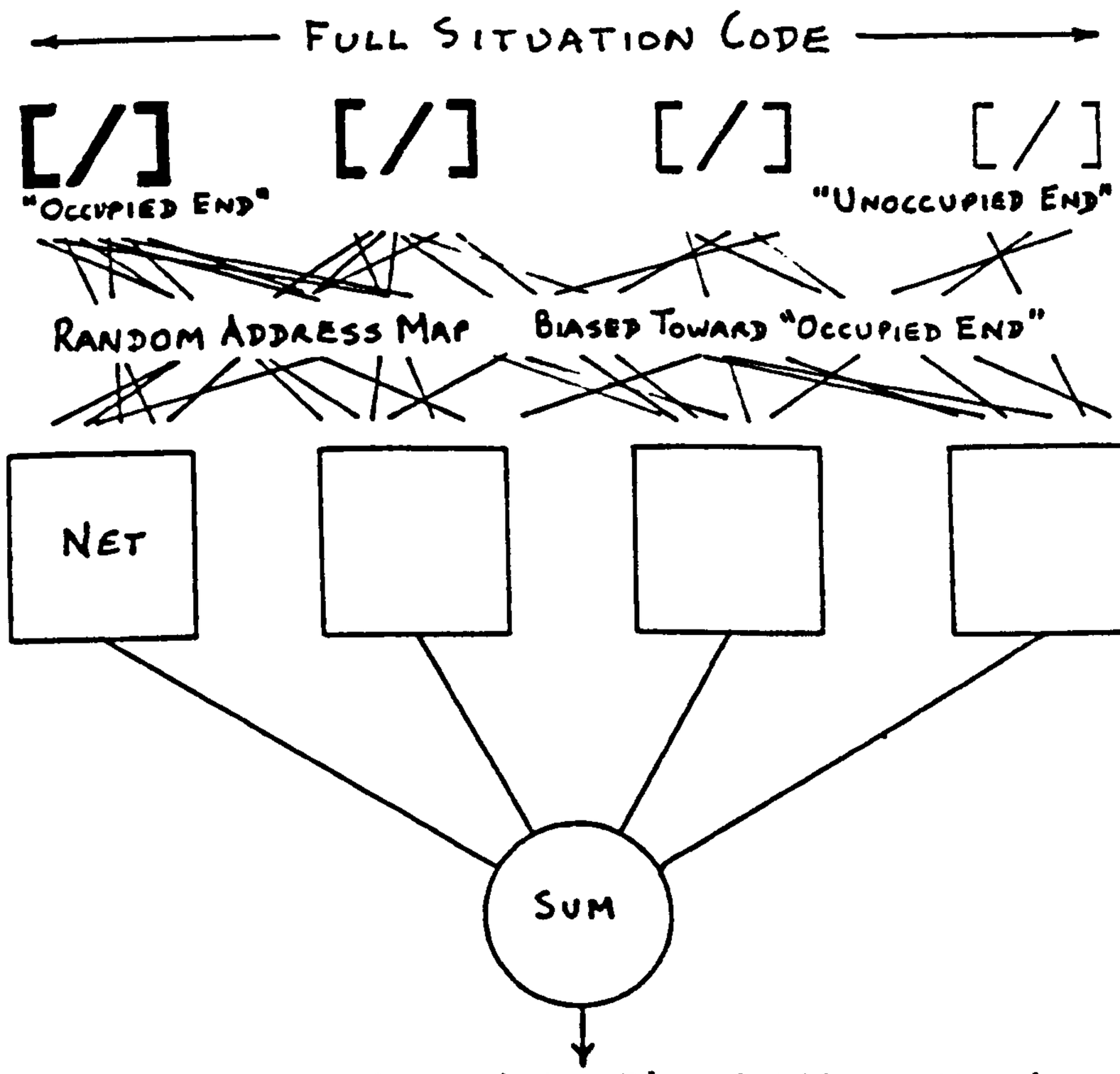


Figure 4.1 Biased Address Mapping

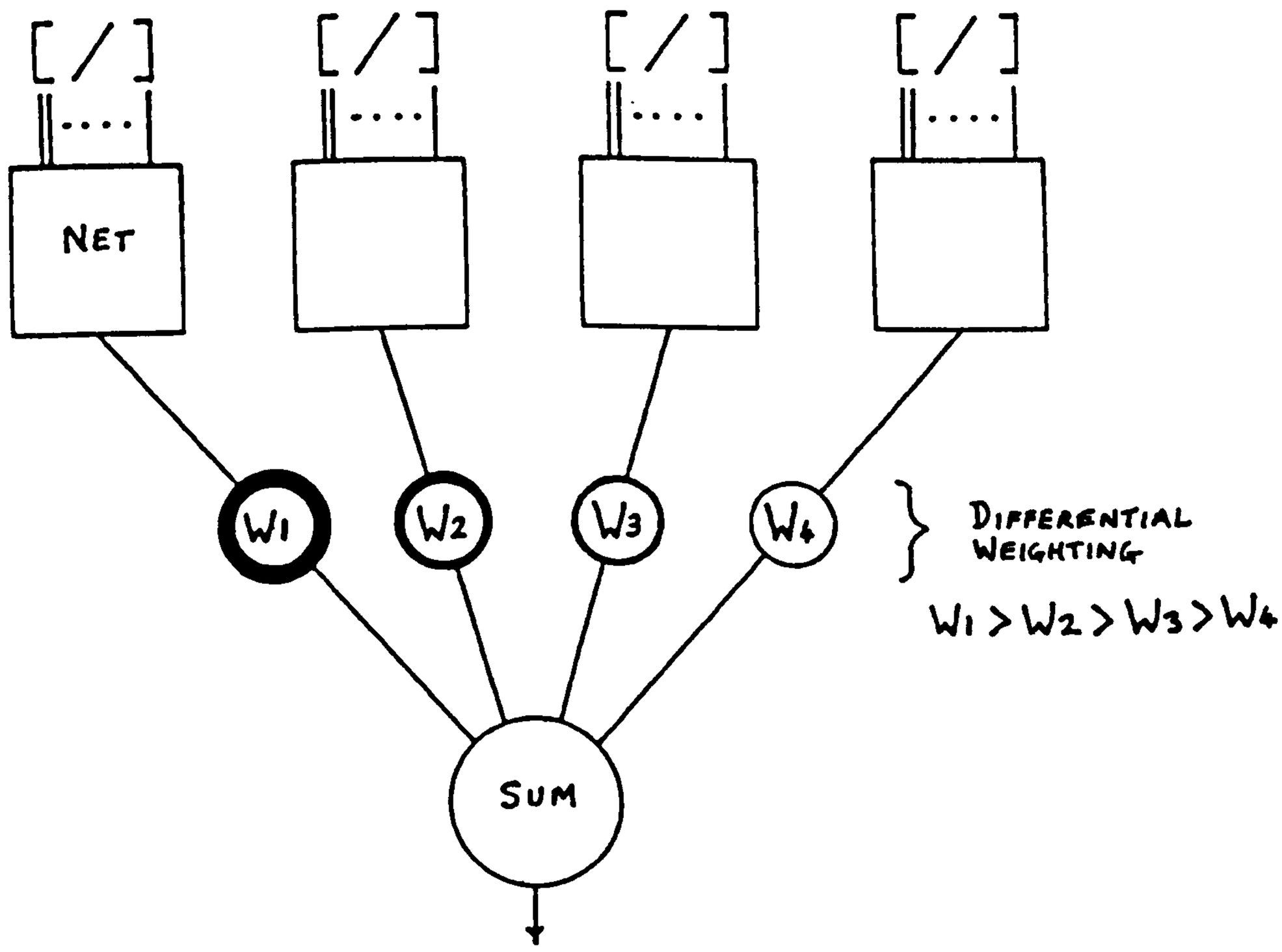


Figure 4.2 Differential Weighting

4.3 Learning Modes

The software beyond System 1 is designed to accommodate three basic modes of training.

a) Human player vs. human opponent with the computer watching and learning from their moves.

b) Computer player vs. human opponent.

c) Computer player vs. computer opponent.

Mode (a) can be used to teach the computer with no prior experience of the game. Mode (b) can follow mode (a) to allow the computer player to test out learned strategies. Finally mode (c) can build up playing experience for the computer player.

Alternatively, each of the modes could be used throughout and the influence of the human element might be discerned from a comparison of the results of using each mode.

4.3.1 Tutor Mode

A combination of modes (b) and (c) was found to be most useful with a human tutor allowing the computer to play a computer opponent but correcting mistakes (to maintain continuity of the game) or giving advice where necessary.

The self-teaching in point (3) of the model implied by the hypothesis of section 3.3.2.3 has particular relevance in this latter (tutor) mode.

The computer player is given the ability to take "imaginary" future moves for both players according to its current best strategy (the contents of the RAMs and settings of parameters). These moves are used to establish the computer player's own assessment of a game situation in terms of where it is likely to lead in future moves of the game (this is explained fully in the hypothesis of section 3.3.2). Self teaching was not implemented.

4.4 Learning in System 2

System 2 is described in section 3.3 of Chapter 3 in relation to the system diagram for System 1 given in fig. 3.1.

In System 2 the learning mode (see section 4.3) is selected by command from the keyboard. With mode (a) two human players play each other using the light pen and tracking grid described in Chapter 2. The computer takes each move to be the best possible and trains the net for an all 1's response at a level given by a human acting as teacher. Thus each move is followed by a request for the "weight" to be associated with that move.

This simple teach action tends to saturate the net in that many discriminators respond equally strongly (with an equal number of 1's). It was refined to allow "detraining" of specified discriminator levels.

In the refined method the teacher specifies either a positive number from 1 to 8 or a negative number from 1 to 8. The positive numbers correspond to the training of a maximum response on that level. The negative numbers indicate that the top n discriminator levels should be detrained to respond with all 0's, where n is the number specified after

the minus sign. The discriminators below the top n are not affected. This allows the teacher to indicate that he requires the move to be valued no higher than a specified level and thus allows detraining of erroneous responses.

Initially the problem of two discriminator levels responding equally was resolved by arbitrarily choosing the one signifying the higher value. Later this problem of saturation of the net was relieved by first detraining all levels (training to give a zero response) before training as before for positive numbers. The negative numbers were retained for corrective action.

Modes (b) and (c) use the same teacher-pupil dialogue as (a). However, the computer first tries to choose a move according to its current best strategy. In the tutor mode (section 4.3.1) the teacher may then change this selection to a better move using the light pen. When the move is confirmed the computer asks for the weight. In all modes the computer asks for a weight after each move only if the learning option is switched on.

Use of the "LEARNING" option in the light pen "menu" allows training to be selectively switched on or off. (The presence on the screen of the "ING" suffix on the end of an option indicates the option is on. The suffix does not appear when an option is selected to be off. See fig. 2.4.)

4.4.1 "Pattern-Background" Attention

Most of the work carried out on the second system uses linear or exponential biasing of the address bit selection for the net.

Address bits to the eight RAMs (common address mapping for all discriminators) are selected, on a randomised basis, such that the number of bits sampled from one end of the move situation code is greater than from the other. The address bits are selected with a linear or exponential bias towards the end of the move situation code reflecting the more heavily occupied lines.

4.4.2 Discussion

The work required to bring System 2 to a stage where useful experimental trials could be done was not considered worthwhile and attention was diverted to System 3. Hence, no experimental results are presented for System 2.

4.5 Learning in System 3

The discussion of training (sections 4.6 and 4.9) and the results (sections 4.7 and 4.8) which follow correspond to the description of System 3 given in section 3.4 of Chapter 3. The text should be read in conjunction with the system diagram in fig. 4.3 below and the search flow chart of fig. 3.12 (explained in section 3.4.4).

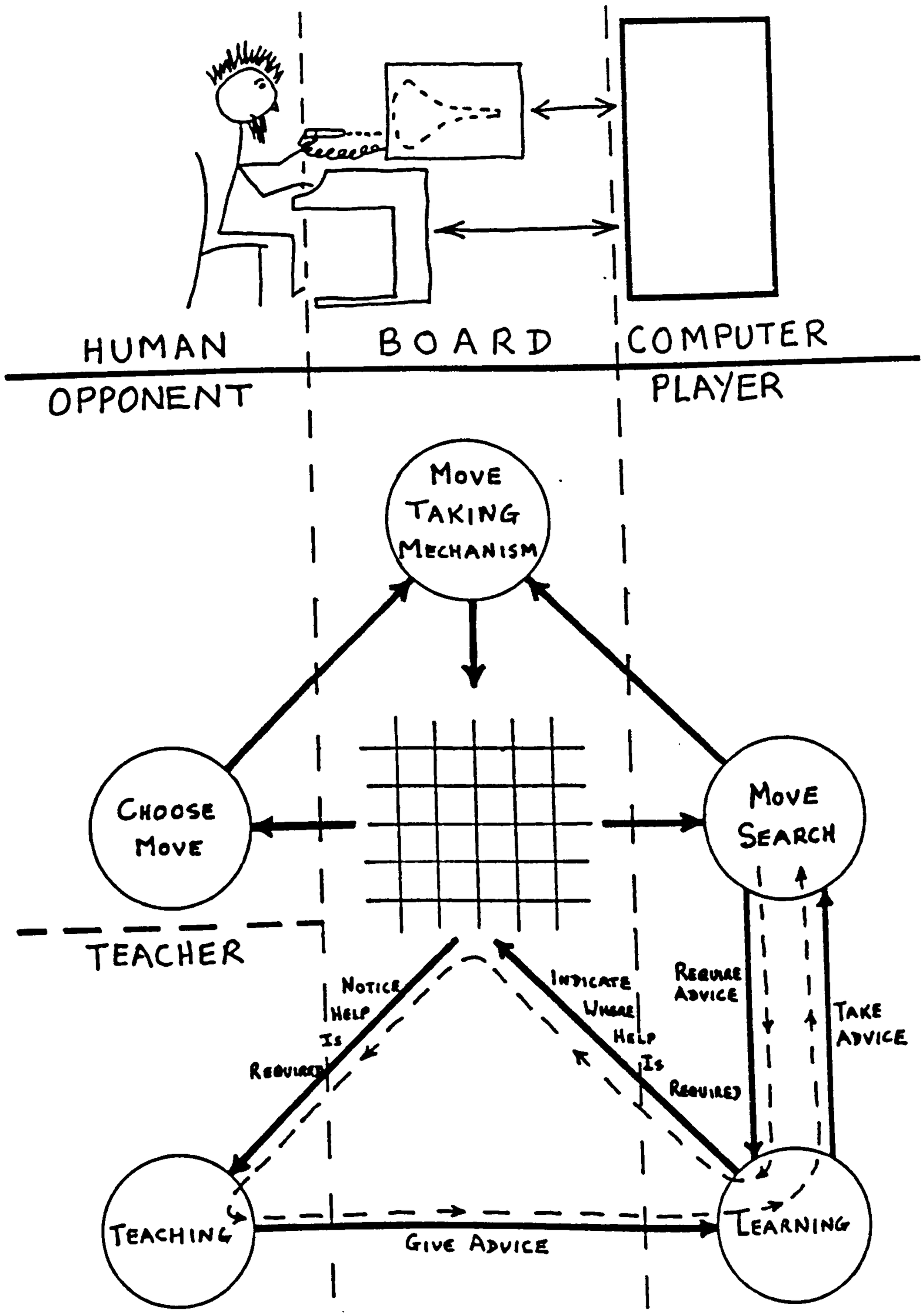


Figure 4.3 Learning Elements of System 3

In the third system, learning is initiated by the computer player when, in learning mode, evaluation of the best line intersecting at a move in the search path indicates that two lines are valued equally. The computer player asks a question of the teacher in order to resolve the difficulty. The teacher then advises the computer player as to which of the two lines should be valued the greater if they are not to be considered equal. The computer player can interpret this advice by adjusting the contents of the simple line evaluation RAM as appropriate, after which the search continues from the point at which it was held up. The additions to the system in fig. 3.8, required for learning, are shown in fig. 4.3.

It is important that advice from the teacher is given in response to a difficulty encountered by (and recognised by) the computer algorithm so that the appropriate question can be asked by the algorithm and it can correctly interpret the teachers advice.

4.6 Training the Line Evaluation Net

During the search path the computer player compares lines intersecting at a move to establish the "best" intersecting line (box 6 in the Move Search flow chart of section 3.4.4, fig. 3.12). Each line is compared, one at a time, with the current best intersecting line. If this comparison indicates the two lines are equally valued then the computer player has no basis for choosing between them.

If, at this point, the computer player is in learning mode, he displays the two line patterns on the graphics screen and asks the teacher to indicate, through the keyboard, which line (if either) is to be valued higher than the other. The teacher can respond by typing the number of

the line (displayed beside each line pattern on the screen) which is to be valued higher, or he types a zero indicating that the lines are to be valued equally.

If the teacher responds with a value other than zero the appropriate line is valued, or "ranked", in future one higher than previously, thus allowing the search algorithm to decide between the two contending lines. If a zero is returned by the teacher the current best intersecting line stands.

The teaching action required in order to value a line pattern one higher than previously is to increment that word of the simple line evaluation RAM indexed by the line pattern (interpreted as a binary number).

The effect of such a training scheme is that lines "compete" for positions (rank) in the valuation scale. Initially all lines are valued equally at zero (the RAM is cleared to zero). Eventually, if the computer is in learning mode, the teacher will be asked to indicate the better of two lines (both valued at zero). The line which he values higher of the two will then have the value one, while all other line patterns will be valued zero. When more than one line pattern has been valued at one the possibility arises that two of these lines will be compared during best line evaluation. If one line is better than the other the teacher's advice will cause it to be valued (one more than previously) at two.

This process of separating unequal line patterns (wrongly valued at the same level) will continue until all line patterns are correctly valued. Thereafter the teacher will always reply that equally valued lines should remain valued equal.

Assuming the teacher gives consistent advice the process of re-ranking will terminate. This is because advice is asked of the teacher only when two lines clash (are ranked, or valued, equal), and a line is only re-ranked by increasing its rank by one. Thus, although the relative ranking of lines can get out of ideal order, any one line can only change rank when it shares rank with another line and by increasing its rank towards its ideal position. Note that this is only the case for consistent teaching and may not therefore apply to self-teaching on the part of the computer player himself.

In the case of advice being asked about two lines which have the same pattern, or when one is the mirror image of the other, the question is not displayed on the screen and the teacher's advice indicating the lines are equal is generated automatically by the program.

The learning behaviour of this learning element (line evaluation net) is illustrated in section 4.7 below.

4.7 Analysis of the Line Evaluation Element

The line evaluation net, or RAM, is referred to as the "line evaluation element" below.

Several analytical devices are presented. The first (Peformance Graphs) treats the learning system as a whole, the remaining analyses look at the line evaluation element in detail. The following sections describe the individual methods of analysis and explain the graphs and tables used with each. The results presented are discussed in section 4.9 on training schemes.

The results given in figs. 4.8, 4.9, 4.10 and 4.11 are presented in six sections corresponding to the six training schemes of fig. 4.21.

These methods of analysis represent a search for a representation of the results which show up any characteristics of the model which might be used to compare different training schemes.

4.7.1 Performance Graphs

This section presents the results obtained from an attempt to "measure" the performance of player models trained in different ways; the measurements being of overall playing performance presented as learning curves for the variously trained players. The graphs of performance (learning curves) show no consistent trend in performance during the training of any one player and cannot therefore form a basis for comparing differently trained players.

The failure of this overall performance approach accounts for the change of emphasis in the remainder of the results presented which concentrate on understanding the behaviour of the playing and learning algorithms at a more detailed level.

The learning curves are presented in fig. 4.4. Each curve is obtained by testing the experience gained by the learning model after each training game. The test involves playing the learning model against a "standard" non-learning model in a ten game match. The learning model under test learns nothing during the test (i.e. LEARNING is turned off). All training and testing games are limited to a maximum of 20 moves for each player in the event of no win. Each point on the graph is obtained

by counting the overall White and Black wins during the ten games, cancelling wins matched by losses for each player. Games which end after 20 moves each, with no win, count as a win for neither player. Thus the graph is obtained by subtracting Black wins from White wins for each point. A positive result gives the number of uncancelled White wins. A negative value gives the uncancelled Black wins.

A more detailed graph constructed on the basis of the number of moves required to win each game was found not to add significantly to the overall win/loose performance graphs of fig. 4.4, so the win/loose graphs were chosen for their greater simplicity.

The three graphs in fig 4.4 correspond to two training schemes:

1) The 'B' and 'W' curves represent the performance of two players trained against each other over 20 training games. Both players have no training before game 1.

2) The 'H' curve represents the performance of a player (White) trained against a non-learning, fully trained opponent (as in fig. 4.14, to be discussed later) during 7 training games.

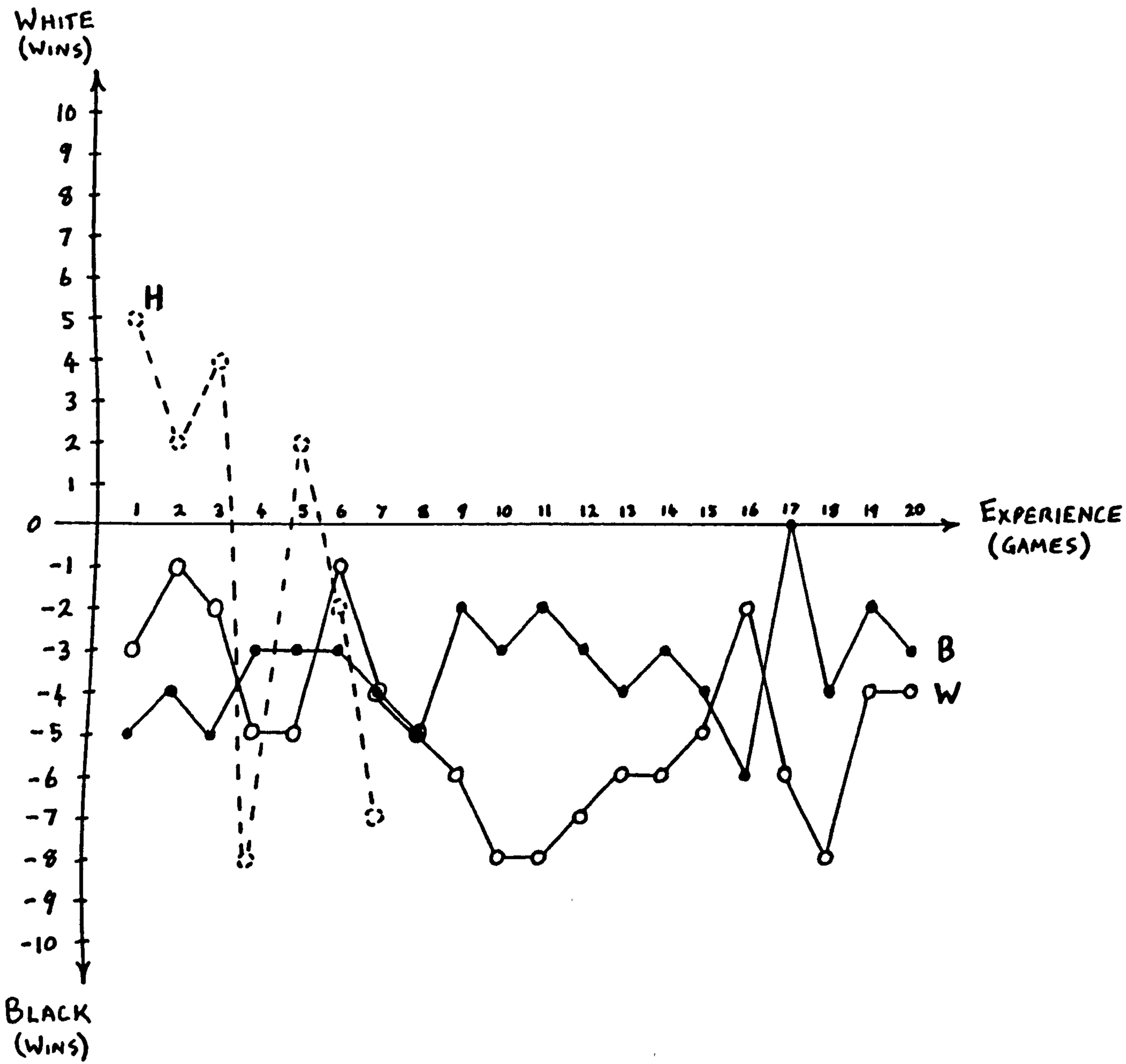


Figure 4.4 Performance Graphs

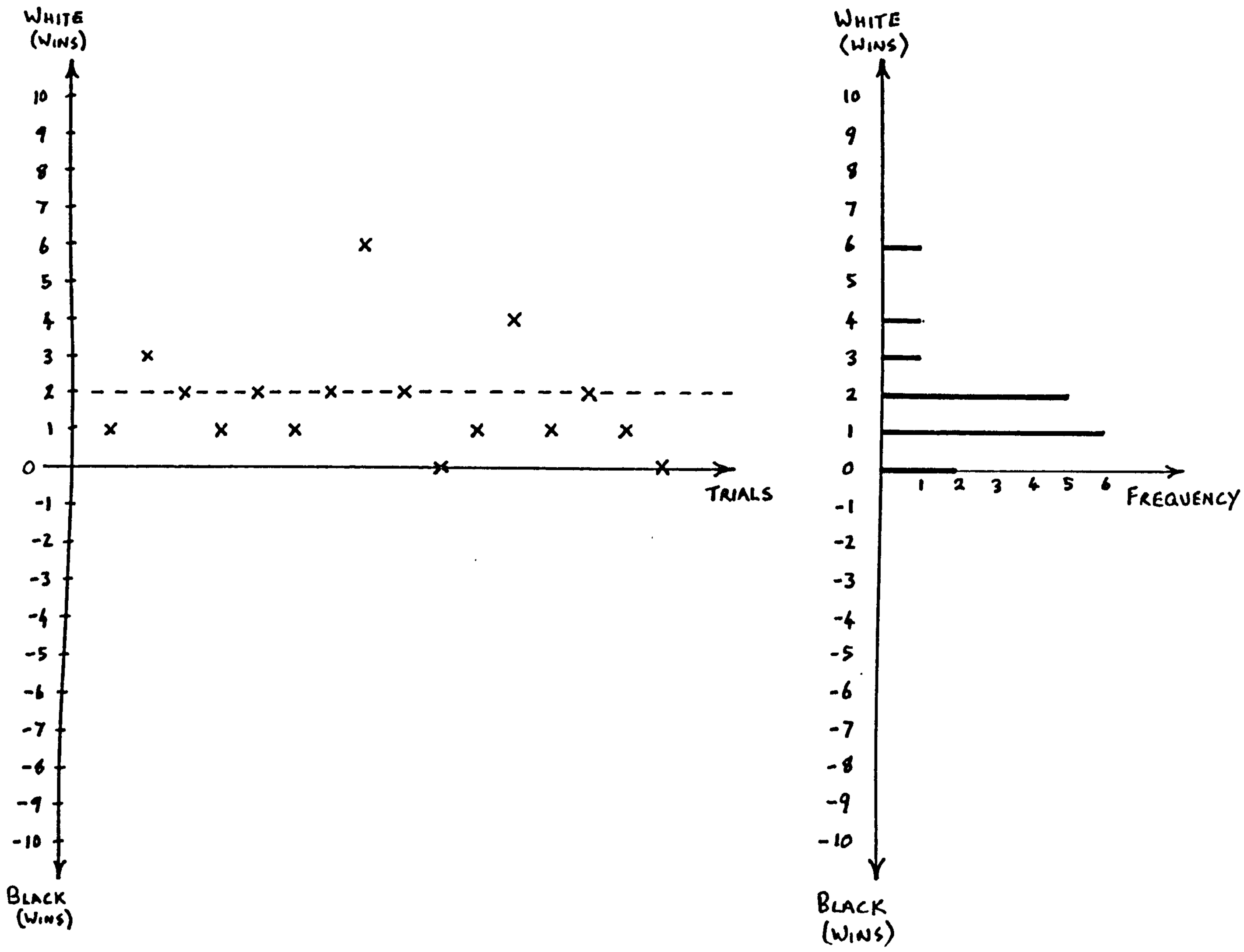


Figure 4.5 Performance Spread

The training for the Peformance Graphs was conducted according to fig. 4.6 below. This training differs from that used for the other experiments in points (a), (c) and (e). See section 4.9 and fig. 4.22 for the later training method.

a) Parameters are set as shown in fig. 4.7 (see section 4.8 for an explanation of parameters.)

b) The initial contents of the line evaluation net are zero (i.e. before game 1).

c) CONFIDENCE is not reset to zero before each game but allowed to vary throughout the training from an initial setting of zero before game 1.

d) Teaching advice is according to the fully trained line evaluation element of fig. 4.14.

e) Training takes place over 20 games (7 games for 'H' curve) with a limit of 20 moves each on any one game.

f) White always takes the first move of each new game.

Figure 4.6 Training Method for Performance Graphs

```
MEDIOCRE=       0
OUTSTANDING=    100

MAXDEADENDS=    3
MAXNODES=       10

LINEPATIENCE=   12
NODEPATIENCE=   12

DON=            0
DOFF=           0
AON=            0
AOFF=           0
```

Figure 4.7 Parameter Settings for Performance Graphs

The results presented here as Performance Graphs are derived from early experiments which accounts for the different training method used in later experiments (section 4.9). The Performance experiments were not repeated with the improved training method used for later experiments because the initial experiments with Performance Graphs are inconclusive due to the lack of clear trends and involve approximately seven and a half minutes computing time for each test point (6 hours for the graphs given in fig. 4.4).

4.7.1.1 Discussion on Performance Graphs

As seen by the performance curves given in fig. 4.4 no significant learning trends are apparent. The method was therefore not used as a means of comparing training methods as planned. Considering fig. 4.5, the "random" appearance of the learning curves in Fig. 4.4 are seen to be a consequence of the wide spread in the values contributing to each point on the learning curves. The points in fig. 4.5 represent overall win values obtained in 16 ten-game tests with the standard fully trained opponent playing against itself. Fig. 4.5 is thus a detailed expansion of a single point in a Performance Graph. The bias towards White wins is due to the advantage of starting first and provides a true zero line for the graphs of fig. 4.4. The overall spread of values can be seen from the hystogram.

Each point on the curves of fig. 4.4 should be viewed in the light of the spread shown in fig. 4.5. The "random" appearance of fig. 4.4 curves indicate the inadequacy of averaging wins over only ten games for each point. The time required to obtain more reliable points based on many more than ten games is not considered worth while.

The remaining experiments represent alternative approaches to presenting and analysing the results of various training schemes. They are discussed below (in sections 4.7.2/3/4/5), after the tables of results in figs. 4.8 to 4.11.

```
XXXXX 000000000000000000000000000001111116666666666666
XXXX- 000000000000000000000000000000000000000000000000
-XXXX 000000000000111111111111111222222222222222222222
XXX-X 000011111111113333333333333333333333333333333333
X-XXX 000000011111112444444444444444445777777777777777
XX-XX 000000000000000000000000000000000000000000000000
-XXX- 000344444444555555555555555555666666666666666666
XXX-- 000000000000000011111111111445555555555555555555
--XXX 000000000022444444444444444445555555555555555555
X-XX- 000000000011111111111111111111111112222222222222
-XX-X 000000001111111111111111111111111111111111111111
XX-X- 000000000011111111111111111111111111111112222222
-X-XX 000000000000000000000000000000000000000000000000
XX--X 000000000000000000000000000000000000000000000000
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 013333333333333333334444444444444444444444444444
--XX- 001233333333333333334444444444444444444444444444
XX--- 002222222222222222223333333333333333333333333333
---XX 000011111112222222222333333333333333333333333333
-X-X- 001111111111111111222222222222222222222222222222
X-X-- 000000000000000000111111111111111111111111111111
--X-X 000000000000000000111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000Q000

      000000000111111111122222222223333333333344444444445
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 124455555555666666666666666666676677772222222222222
XXXX- 124455555555666666666666666666677788888888888888888
-XXXX 124455555555655555555555555554455566666666666666666
XXX-X 124444444445533333333333333334445555555555555555555
X-XXX 124455544445554222222222222223321111111111111111111
XX-XX 124455555555666666666666666666677788888888888888888
-XXX- 124111111111111111111111111111111122222222222222222
XXX-- 124455555555666665555555555555222223333333333333333
--XXX 124455555533222222222222222222232233333333333333333
X-XX- 124455555554555555555555555555566677776666666666666
-XX-X 124455554445555555555555555555566677777777777777777
XX-X- 124455555555555555555555555555566677777777777666666
-X-XX 124455555555666666666666666666677788888888888888888
XX--X 124455555555666666666666666666677788888888888888888
X--XX 124455555555666666666666666666677788888888888888888
X-X-X 124455555555666666666666666666677788888888888888888
-XX-- 111122222222333333333322222222333344444444444444444
--XX- 123222222222333333333322222222333344444444444444444
XX--- 122233333333444444444433333333444555555555555555555
---XX 124444444443444444444433333333444555555555555555555
-X-X- 123344444444555555444444444444455566666666666666666
X-X-- 124455555555666666555555555555566677777777777777777
--X-X 124455555555666666555555555555566677777777777777777
X--X- 124455555555666666666666666666677788888888888888888
-X--X 124455555555666666666666666666677788888888888888888
X---X 124455555555666666666666666666677788888888888888888
--X-- 124455555555666666666666666666677788888888888888888
-X--- 124455555555666666666666666666677788888888888888888
---X- 124455555555666666666666666666677788888888888888888
X---- 124455555555666666666666666666677788888888888888888
----X 124455555555666666666666666666677788888888888888888
----- 124455555555666666666666666666677788888888888888888

      000000000111111111122222222223333333333344444444445
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(1)a Line Value Tabulations (WA)

```
XXXXX 6666666666666666667777777777777777777777777777777
XXXX- 0000000000000000001111111111111111111222222222222
-XXXX 2222222222222222222222222222222222222222222222222
XXX-X 3333333333333333333333333333333333333333333333333
X-XXX 7777777777777777778888888888888888888888888888888
XX--XX 0000000000000000000000000000000000000000000000000
-XXX- 6666666666666666666777777777777777777777777777777
XXX-- 5555555555555555555555556666666666666666666666666
--XXX 5555555555555555555556666666666666666666666666666
X-XX- 2222222222222222222222222222222222222222222222222
-XX-X 1111111111122222222222222222223333333333333333333
XX-X- 2222222222222222455555555555555555555555555555555
-X-XX 0000000000000000000000000000000000000000000000000
XX--X 0000000000000000000000000000000000000000000000000
X---XX 0000000000000000000000000000000000000000000000000
X-X-X 0000000000000000000000000000000000000000000000000
-XX-- 4444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444
XX--- 3333333333333333333333333333333333333333333333333
---XX 3333333333333333333333333333333333333333333333333
-X-X- 2222222222222222222222222222222222222222222222222
X-X-- 1111111111111111111111111111111111111111111111111
--X-X 1111111111111111111111111111111111111111111111111
X--X- 0000000000000000000000000000000000000000000000000
-X--X 0000000000000000000000000000000000000000000000000
X---X 0000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000

      0000000000000000000000000000000000000000000000001
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789 0
```

```
XXXXX 2222222222222222221222222222222222222222222222222
XXXX- 8888888888888888888788888888888888888888777777777
-XXXX 6666666666666666666777777777777777777777777777777
XXX-X 5555555555555555555666666666666666666666666666666
X-XXX 1111111111111111111111111111111111111111111111111
XX-XX 8888888888888888888999999999999999999999999999999
-XXX- 2222222222222222222222222222222222222222222222222
XXX-- 3333333333333333333444433333333333333333333333333
--XXX 3333333333333333333233333333333333333333333333333
X-XX- 6666666666666666666777777777777777777777777777777
-XX-X 7777777777766666666677777777777766666666666666666
XX-X- 6666666666666666433344444444444444444444444444444
-X-XX 8888888888888888888999999999999999999999999999999
XX--X 8888888888888888888999999999999999999999999999999
X--XX 8888888888888888888999999999999999999999999999999
X-X-X 8888888888888888888999999999999999999999999999999
-XX-- 4444444444444444444333333333333333333333333333333
--XX- 4444444444444444444333333333333333333333333333333
XX--- 5555555555555555555666666666666666666666666666666
---XX 5555555555555555555666666666666666666666666666666
-X-X- 6666666666666666666777777777777777777777777777777
X-X-- 7777777777777777778888888888888888888888888888888
--X-X 7777777777777777778888888888888888888888888888888
X--X- 8888888888888888888999999999999999999999999999999
-X--X 8888888888888888888999999999999999999999999999999
X---X 8888888888888888888999999999999999999999999999999
--X-- 8888888888888888888999999999999999999999999999999
-X--- 8888888888888888888999999999999999999999999999999
---X- 8888888888888888888999999999999999999999999999999
X---- 8888888888888888888999999999999999999999999999999
----X 8888888888888888888999999999999999999999999999999
----- 8888888888888888888999999999999999999999999999999

      0000000000000000000000000000000000000000000000001
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789 0
```

Figure 4.8(1)b continued

PAGE NUMBERS CUT OFF IN ORIGINAL

```
XXXXX  77777777777777777777777777777777777777778888888888
XXXX-  22222222222222222222222222222222222222222222222222
-XXXX  22222222222222222222224444444444444777777777777777
XXX-X  33333333333333333333333333333333333333333333333333
X-XXX  00000000000000000000000000000000000000000000000000
XX-XX  00000000000000000000000000000000000000000000000000
-XXX-  77777777777777777777777777777777777777777777777777
XXX--  66666666666666666666666666666666666666666666666666
--XXX  66666666666666666666666666666666666666666666666666
X-XX-  22222222222222222222222222222222222222222222222222
-XX-X  33333335555555555555555556666666666666666666666666
XX-X-  55555555555555555555555555555555555555555555555555
-X-XX  00000000000000000000000000000000000000000000000000
XX--X  00000000000000000000000000000000000000000000000000
X--XX  00000000000000000000000000000000000000000000000000
X-X-X  00000000000000000000000000000000000000000000000000
-XX--  44444444444444444444444444444444444444444444444444
--XX-  44444444444444444444444444444444444444444444444444
XX---  33333333333333333333333333333333333333333333333333
---XX  33333333333333333333333333333333333333333333333333
-X-X-  22222222222222222222222222222222222222222222222222
X-X--  11111111111111111111111111111111111111111111111111
--X-X  11111111111111111111111111111111111111111111111111
X--X-  00000000000000000000000000000000000000000000000000
-X--X  00000000000000000000000000000000000001111111111111
X---X  00000000000000000000000000000000000000000000000000
--X--  00000000000000000000000000000000000000000000000000
-X---  00000000000000000000000000000000000000000000000000
---X-  00000000000000000000000000000000000000000000000000
X----  00000000000000000000000000000000000000000000000000
----X  00000000000000000000000000000000000000000000000000
-----  00000000000000000000000000000000000000000000000000

       11111111111111111111111111111111111111111111111111
       00000000011111111112222222222333333333344444444445
       12345678901234567890123456789012345678901234567890
```

WA.140

```
XXXXX  22222222222222222222222222222222222222221111111111
XXXX-  77777777777777777777777777777777777777777777777777
-XXXX  77777777777777777777777755555555555555222222222222
XXX-X  66666666666666666666666666666666666666666666666666
X-XXX  11111111111111111111111111111111111111111111111111
XX-XX  99999999999999999999999999999999999999999999999999
-XXX-  22222222222222222222222222222222222222222222222222
XXX--  33333333333333333333333333333333333333333333333333
--XXX  33333333333333333333333333333333333333333333333333
X-XX-  77777777777777777777777777777777777777777777777777
-XX-X  66666644444444444444444444333333333333333333333333
XX-X-  44444444444444444444444444444444444444444444444444
-X-XX  99999999999999999999999999999999999999999999999999
XX--X  99999999999999999999999999999999999999999999999999
X--XX  99999999999999999999999999999999999999999999999999
X-X-X  99999999999999999999999999999999999999999999999999
-XX--  55555555555555555555555555555555555555555555555555
--XX-  55555555555555555555555555555555555555555555555555
XX---  66666666666666666666666666666666666666666666666666
---XX  66666666666666666666666666666666666666666666666666
-X-X-  77777777777777777777777777777777777777777777777777
X-X--  88888888888888888888888888888888888888888888888888
--X-X  88888888888888888888888888888888888888888888888888
X--X-  99999999999999999999999999999999999999999999999999
-X--X  99999999999999999999999999999999998888888888888888
X---X  99999999999999999999999999999999999999999999999999
--X--  99999999999999999999999999999999999999999999999999
-X---  99999999999999999999999999999999999999999999999999
---X-  99999999999999999999999999999999999999999999999999
X----  99999999999999999999999999999999999999999999999999
----X  99999999999999999999999999999999999999999999999999
-----  99999999999999999999999999999999999999999999999999

       11111111111111111111111111111111111111111111111111
       00000000011111111112222222222333333333344444444445
       12345678901234567890123456789012345678901234567890
```

Figure 4.8(1)c continued

```
XXXXX 0000000000000000000000000000000000000000000000000000000
XXXX- 2222222222222222222222222222222222222222222222222222222
XXX-- 4444444444444444444444444444444444444444444444444444444
-XXXX 4444444444444444444444444444444444444444444444444444444
XXX-X 3333333333333333333333333333333333333333333333333333333
X-XXX 8888888888888888888888888888888888888888888888888888888
XX-XX 0000000000000000000000000000000000000000000000000000000
-XXX- 7777777777777777777777777777777777777777777777777777777
XXX-- 6666666666666666666666666666666666666666666666666666666
--XXX 6666666666666666666666666666666666666666666666666666666
X-XX- 2222222222222222222222222222222222222222333333333333333
-XX-X 6666666666666666666666666666666666666666666666666666666
XX-X- 5555555555555555555555555555555555555555555555555555555
-X-XX 0000000000000000000000000000000000000000000000000000000
XX--X 0000000000000000000000000000000000000000000000000000000
X---XX 0000000000000000000000000000000000000000000000000000000
X-X-X 0000000000000000000000000000000000000000000000000000000
-XX-- 4444444444444444444444444444444444444444444444555555555
--XX- 4444444444444444444444444444444444444444444444555555555
XX--- 3333333333333333333333333333333333333333333333444444444
---XX 3333333333333333333333333333333333333333333333444444444
-X-X- 2222222222222222222222222222333333333333333333333333333
X-X-- 1111111111111111111111111111111111111111111111111111111
--X-X 1111111111111111111111111111222222222222222222222222222
X---X- 0000000000000000000000000000000000000000000000000000000
-X--X 1111111111111111111111111111111111111111111111111111111
X---X 0000000000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000000000

      1111111111111111111111111111111111111111111111111111112
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012345678901234567890
```

```
XXXXX 1111111111111111111111111111111111111111111111111111111
XXXX- 7777777777777777777777777777777777777777777777777777777
-XXXX 2222222222222222222222222222222222222222222222222222222
XXX-X 6666666666666666666666666666666666666666666666666666666
X-XXX 1111111111111111111111111111111111111111111111111111111
XX-XX 9999999999999999999999999999999999999999999999999999999
-XXX- 2222222222222222222222222222222222222222222222222222222
XXX-- 3333333333333333333333333333333333333333333333333333333
--XXX 3333333333333333333333333333333333333333333333333333333
X-XX- 7777777777777777777777777777777777777777666666666666666
-XX-X 3333333333333333333333333333333333333333333333333333333
XX-X- 4444444444444444444444444444444444444444444444444444444
-X-XX 9999999999999999999999999999999999999999999999999999999
XX--X 9999999999999999999999999999999999999999999999999999999
X--XX 9999999999999999999999999999999999999999999999999999999
X-X-X 9999999999999999999999999999999999999999999999999999999
-XX-- 5555555555555555555555555555555555555555555555555544444
--XX- 5555555555555555555555555555555555555555555555555544444
XX--- 6666666666666666666666666666666666666666666666665555555
---XX 6666666666666666666666666666666666666666666666665555555
-X-X- 7777777777777777777777777777666666666666666666666666666
X-X-- 8888888888888888888888888888888888888888888888888888888
--X-X 8888888888888888888888888887777777777777777777777777777
X--X- 9999999999999999999999999999999999999999999999999999999
-X--X 8888888888888888888888888888888888888888888888888888888
X---X 9999999999999999999999999999999999999999999999999999999
--X-- 9999999999999999999999999999999999999999999999999999999
-X--- 9999999999999999999999999999999999999999999999999999999
---X- 9999999999999999999999999999999999999999999999999999999
X---- 9999999999999999999999999999999999999999999999999999999
----X 9999999999999999999999999999999999999999999999999999999
----- 9999999999999999999999999999999999999999999999999999999

      1111111111111111111111111111111111111111111111111111112
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012345678901234567890
```

Figure 4.8(1)d continued

```
XXXXX 000000000000000000000000000000000000000000000000000
XXXX- 000000000000000000000000000000000000000000000000000
-XXXX 000000000000000000000000000000000000000000000000000
XXX-X 000000000000000000000000000000022222222222222222222
X-XXX 000000000000000000000000000000000000000000011111222
XX-XX 000000000000000000000000000000000000000000000000000
-XXX- 000000011111113344444444444444444444444444444444444
XXX-- 000000000000000000011111111111111111111111111111111
--XXX 000000011111111111111111111111111111111111111111111
X-XX- 000000000000000000011111111111122233333333344444444
-XX-X 000000011111111111222222222222222222222222222222222
XX-X- 000011111111111111111111111111111111111111111111111
-X-XX 000000000000000000000000000000000011111111111111111
XX--X 000000000000000000000000000000000000000000000000000
X--XX 000000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000000
-XX-- 033333333333333333333333333333333333333333333333333
--XX- 011111111111111111111233333333333333333333333333333
XX--- 000000000022222222222222222222222222222222222222222
---XX 022222222222222222222222222222222222222222222222222
-X-X- 011111111111111111111111111111111111111111111111111
X-X-- 000000000000000000000000000000000000000000000000000
--X-X 000000000000000000000000000000000000000000000000000
X--X- 000000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000000

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 144444444444444555555555555555555555555555555555555
XXXX- 144444444444444555555555555555555555555555555555555
-XXXX 144444444444444555555555555555555555555555555555555
XXX-X 144444444444444555555555555555555555333333333333333
X-XXX 144444444444444555555555555555555555555555444443333
XX-XX 144444444444444555555555555555555555555555555555555
-XXX- 144444433333311111111111111111111111111111111111111
XXX-- 144444444444444555554444444444444444444444444444444
--XXX 144444433333333444444444444444444444444444444444444
X-XX- 144444444444444555554444444444444333222222221111111
-XX-X 144444433333333444433333333333333333333333333333333
XX-X- 144433333333333444444444444444444444444444444444444
-X-XX 144444444444444555555555555555555555444444444444444
XX--X 144444444444444555555555555555555555555555555555555
X--XX 144444444444444555555555555555555555555555555555555
X-X-X 144444444444444555555555555555555555555555555555555
-XX-- 111111111111111222222222222222222222222222222222222
--XX- 133333333333333444444322222222222222222222222222222
XX--- 144444444422222333333333333333333333333333333333333
---XX 122222222222222333333333333333333333333333333333333
-X-X- 133333333333333444444444444444444444444444444444444
X-X-- 144444444444444555555555555555555555555555555555555
--X-X 144444444444444555555555555555555555555555555555555
X--X- 144444444444444555555555555555555555555555555555555
-X--X 144444444444444555555555555555555555555555555555555
X---X 144444444444444555555555555555555555555555555555555
--X-- 144444444444444555555555555555555555555555555555555
-X--- 144444444444444555555555555555555555555555555555555
---X- 144444444444444555555555555555555555555555555555555
X---- 144444444444444555555555555555555555555555555555555
----X 144444444444444555555555555555555555555555555555555
----- 144444444444444555555555555555555555555555555555555

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(2)a Line Value Tabulations (W6)

```
XXXXX 000000000000000000000000000000000000000000000000000
XXXX- 000000000000000000111111111111111111111111111111111
-XXXX 000000000000000000000000000000000000000000000000000
XXX-X 222222222222222222222222222222222222222222222222222
X-XXX 222222222222222222222222222222224444444444444555555
XX-XX 000000000000000000000000000000000000000000000000000
-XXX- 444444444444444444444444444444444444444444444444455
XXX-- 111111111111111111111111111111111112222222222222222
--XXX 111111111111111111111111111111111111111111111111111
X-XX- 444444444444444444444444444444555556666666666666666
-XX-X 222222222222222222222222222222222222222222222222222
XX-X- 111111111111111111111111111111111111111111111111111
-X-XX 111111111111111111111112233333333335555555555555555
XX--X 000000000000000000000000000000000000000000000000000
X--XX 000000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000000
-XX-- 333333333333333333333333334444444444444444444444444
--XX- 333333333333333333333333334444444444444444444444444
XX--- 222222222222222222222222222333333333333333333333333
---XX 222222222222222222222222223333333333333333333333333
-X-X- 111111111111111111111111112222222222222222222222222
X-X-- 000000000000000000000000000000000000000000000000000
--X-X 000000000000000000111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000000

      000000000000000000000000000000000000000000000000001
      555555555566666666667777777777888888888899999999990
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 555555555555555555555555555555555555666677777777777777777
XXXX- 555555555555555555555555444444444445555666666666666666666
-XXXX 555555555555555555555555555555555555666677777777777777777
XXX-X 333333333333333333333333333333333344445555555555555555555
X-XXX 333333333333333333333333333333333444433333333333333333222
XX-XX 555555555555555555555555555555555555666677777777777777777
-XXX- 111111111111111111111111111111111112222333333333333333322
XXX-- 444444444444444444444444444444444445555555555555555555555
--XXX 444444444444444444444444444444444445555666666666666666666
X-XX- 111111111111111111111111111111111111111111111111111111111
-XX-X 333333333333333333333333333333333344443555555555555555555
XX-X- 444444444444444444444444444444444445555666666666666666666
-X-XX 444444444444444444444444443332222233333222222222222222222
XX--X 555555555555555555555555555555555555666677777777777777777
X--XX 555555555555555555555555555555555555666677777777777777777
X-X-X 555555555555555555555555555555555555666677777777777777777
-XX-- 222222222222222222222222222222211111222223333333333333333
--XX- 222222222222222222222222222222211111222223333333333333333
XX--- 333333333333333333333333333333332233334444444444444444444
---XX 333333333333333333333333333333222223333344444444444444444
-X-X- 444444444444444444444444443333333444455555555555555555555
X-X-- 555555555555555555555555555555555555666677777777777777777
--X-X 555555555555555555555555444444444445555666666666666666666
X--X- 555555555555555555555555555555555555666677777777777777777
-X--X 555555555555555555555555555555555555666677777777777777777
X---X 555555555555555555555555555555555555666677777777777777777
--X-- 555555555555555555555555555555555555666677777777777777777
-X--- 555555555555555555555555555555555555666677777777777777777
---X- 555555555555555555555555555555555555666677777777777777777
X---- 555555555555555555555555555555555555666677777777777777777
----X 555555555555555555555555555555555555666677777777777777777
----- 555555555555555555555555555555555555666677777777777777777

      000000000000000000000000000000000000000000000000001
      555555555566666666667777777777888888888899999999990
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(2)b continued

```
XXXXX 0000000000000000000000000000000000000000000000000
XXXX- 1111111111111111111111111111111111111111111111111
-XXXX 0000000000000000000000000000000000000000000000000
XXX-X 2222222222222222222222222222222222222222222222222
X-XXX 5555555555555555666666666666666666666666667777
XX-XX 0000000000000000000000000000000000000000000000000
-XXX- 5555566666666666666666666666666666666666666666666
XXX-- 2222222222222222222222222222222222222222222222222
--XXX 1111111111111111111112222222222222222222222222222
X-XX- 6666666666666666666666666666666666666666666666666
-XX-X 2222222222222222222222222222222222222222222222222
XX-X- 1111111111111111111111111111111111111111111111111
-X-XX 5555555555555555555555555555555555555555555555555
XX--X 0000000000000000000000000000000000000000000000000
X--XX 0000000000000000000000000000000000000000000000000
X-X-X 0000000000000000000000000000000000000000000000000
-XX-- 4444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444
XX--- 3333333333333333333333333333333333333333333333333
---XX 3333333333333333333333333333333333333333333333333
-X-X- 2222222222222222222222222222222222222222222222222
X-X-- 0000000000000001111111111111111111111111111111111
--X-X 1111111111111111111111111111111111111111111111111
X--X- 0000000000000000000000000000000000000000000000000
-X--X 0000000000000000000000000000000000000000000000000
X---X 0000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000

      1111111111111111111111111111111111111111111111111
      0000000001111111111222222222233333333334444444445
      1234567890123456789012345678901234567890123456789 0
```

```
XXXXX 7777777777777777777777777777777777777777777778888
XXXX- 6666666666666666666666666666666666666666667777
-XXXX 7777777777777777777777777777777777777777778888
XXX-X 5555555555555555555555555555555555555555556666
X-XXX 2222222222222221111111111111111111111111111111
XX-XX 7777777777777777777777777777777777777777778888
-XXX- 2222211111111111111111111111111111111111112222
XXX-- 5555555555555555555555555555555555555555556666
--XXX 6666666666666666666665555555555555555555556666
X-XX- 1111111111111111111111111111111111111111112222
-XX-X 5555555555555555555555555555555555555555556666
XX-X- 6666666666666666666666666666666666666666667777
-X-XX 2222222222222222222222222222222222222222333333
XX--X 7777777777777777777777777777777777777777778888
X--XX 7777777777777777777777777777777777777777778888
X-X-X 7777777777777777777777777777777777777777778888
-XX-- 3333333333333333333333333333333333333333334444
--XX- 3333333333333333333333333333333333333333334444
XX--- 4444444444444444444444444444444444444444445555
---XX 4444444444444444444444444444444444444444445555
-X-X- 5555555555555555555555555555555555555555556666
X-X-- 7777777777777776666666666666666666666666667777
--X-X 6666666666666666666666666666666666666666667777
X--X- 7777777777777777777777777777777777777777778888
-X--X 7777777777777777777777777777777777777777778888
X---X 7777777777777777777777777777777777777777778888
--X-- 7777777777777777777777777777777777777777778888
-X--- 7777777777777777777777777777777777777777778888
---X- 7777777777777777777777777777777777777777778888
X---- 7777777777777777777777777777777777777777778888
----X 7777777777777777777777777777777777777777778888
----- 7777777777777777777777777777777777777777778888

      1111111111111111111111111111111111111111111111111
      0000000001111111111222222222233333333334444444445
      1234567890123456789012345678901234567890123456789 0
```

Figure 4.8(2)c continued

```
XXXXX 0000000000000000000000000000000000002222222222223
XXXX- 1111111111111111111111111111111111111111111111111
-XXXX 0000000000000000000000000000000000000000000000000
XXX-X 2222222222222222222222222222222222222222222222222
X-XXX 7777777777777777777777777777777777777777777777777
XX-XX 0000000000000000000000000000000000000000000000000
-XXX- 6666666666666666666666666666666666666666666666666
XXX-- 2222222222222222222222222222222222222222222222222
--XXX 2222222222222222445555555555555555555555555555555
X-XX- 6666666666666666666666666666666666666666666666666
-XX-X 2222222222222222222222222222222222222222222222222
XX-X- 1111111111111111111111111111111333333333333333555
-X-XX 5555555555555555555555555555555555555555555555555
XX--X 0000000000000000000000000000000000000000000000000
X--XX 0000000000000000000000000000000000000000000000000
X-X-X 0000000000000000000000000000000000000000000000000
-XX-- 4444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444
XX--- 3333333333333333333333333333333333333333333333333
---XX 3333333333333333333333333333333333333333333333333
-X-X- 2222222222222222222222222222222222222222222222222
X-X-- 1111111111111111111111111111111111111111111111111
--X-X 1111111111111111111111111111111111111111111111111
X--X- 0000000000000000000000000000000000000000000000000
-X--X 0000000000000001111111111111111111111111111111111
X---X 0000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000

      1111111111111111111111111111111111111111111111112
      5555555555666666666677777777778888888888999999999 0
      1234567890123456789012345678901234567890123456789 0
```

```
XXXXX 8888888888888888888888888888888888888866666666666665
XXXX- 7777777777777777777777777777777777777777777777777777
-XXXX 8888888888888888888888888888888888888888888888888888
XXX-X 6666666666666666666666666666666666666666666666666666
X-XXX 1111111111111111111111111111111111111111111111111111
XX-XX 9999999999999999999999999999999999999999999999999999
-XXX- 2222222222222222222222222222222222222222222222222222
XXX-- 6666666666666666666666666666666666666666666666666666
--XXX 6666666666666644333333333333333333333333333333333333
X-XX- 2222222222222222222222222222222222222222222222222222
-XX-X 6666666666666666666666666666666666666666666666666666
XX-X- 7777777777777777777777777777755555555555555333333333
-X-XX 3333333333333333333333333333333333333333333333333333
XX--X 8888888888888888888888888888888888888888888888888888
X--XX 8888888888888888888888888888888888888888888888888888
X-X-X 8888888888888888888888888888888888888888888888888888
-XX-- 4444444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444444
XX--- 5555555555555555555555555555555555555555555555555555
---XX 5555555555555555555555555555555555555555555555555555
-X-X- 6666666666666666666666666666666666666666666666666666
X-X-- 7777777777777777777777777777777777777777777777777777
--X-X 7777777777777777777777777777777777777777777777777777
X--X- 8888888888888888888888888888888888888888888888888888
-X--X 8888888888888887777777777777777777777777777777777777
X---X 8888888888888888888888888888888888888888888888888888
--X-- 8888888888888888888888888888888888888888888888888888
-X--- 8888888888888888888888888888888888888888888888888888
---X- 8888888888888888888888888888888888888888888888888888
X---- 8888888888888888888888888888888888888888888888888888
----X 8888888888888888888888888888888888888888888888888888
----- 8888888888888888888888888888888888888888888888888888

      1111111111111111111111111111111111111111111111112
      5555555555666666666677777777778888888888999999999 0
      1234567890123456789012345678901234567890123456789 0
```

Figure 4.8(2)d continued

```
XXXXX 000000000000000000000000000000000000000000111111
XXXX- 000000000000000000000000000000000000000002222222
-XXXX 000000111122222333333555556666666666666666666666
XXX-X 000001111111111111111111111111111111111111111111
X-XXX 000000111112222222222222222222222222222222222222
XX-XX 000000000000000000000000000000000000000000000000
-XXX- 000000000000000000000011111112222234444444444444
XXX-- 000000000000000000000002555555555555555555555555
--XXX 000333444444444444444444445555555555555555555555
X-XX- 000111111111111111111111222222222222222222222222
-XX-X 000000000000000000000000000000000000000000000000
XX-X- 000000000000000000000000000000000000000000000000
-X-XX 000000000011111111111133333335555555555555555555
XX--X 000000000000022222222222222222222222222222222222
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 022233333333333333333334444444444444444444444444
--XX- 022233333333333333333334444444444444444444444444
XX--- 011122222222222222222222222222222222222222222233
---XX 011111111112222222222223333333333333333333333333
-X-X- 000011111111111111111112222222222222222222222222
X-X-- 000000000000000000000000000000000000000000000000
--X-X 000000000000000000001111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 133444555555555555555556666677777777777777777666666
XXXX- 133444555555555555555556666677777777777777777555555
-XXXX 133444444433333222222211111111111111111111111111111
XXX-X 133443444444444444444445555566666666666666666666666
X-XXX 133444444443333333333344444555555555555555555555555
XX-XX 133444555555555555555556666677777777777777777777777
-XXX- 133444555555555555555565555566655555433333333333333
XXX-- 133444555555555555555556666412222222222222222222222
--XXX 133111111111111111111112222122222222222222222222222
X-XX- 133333444444444444444445444455555555555555555555555
-XX-X 133444555555555555555556666677777777777777777777777
XX-X- 133444555555555555555556666677777777777777777777777
-X-XX 133444555544444444444443333344222222222222222222222
XX--X 133444555555533333333334444455555555555555555555555
X--XX 133444555555555555555556666677777777777777777777777
X-X-X 133444555555555555555556666677777777777777777777777
-XX-- 111211222222222222222233222333333333333333333333333
--XX- 111211222222222222222233222333333333333333333333333
XX--- 122322333333333333333344444555555555555555555555544
---XX 122333444443333333333344433344444444444444444444444
-X-X- 133433444444444444444445444455555555555555555555555
X-X-- 133444555555555555555556666677777777777777777777777
--X-X 133444555555555555555544555556666666666666666666666
X--X- 133444555555555555555556666677777777777777777777777
-X--X 133444555555555555555556666677777777777777777777777
X---X 133444555555555555555556666677777777777777777777777
--X-- 133444555555555555555556666677777777777777777777777
-X--- 133444555555555555555556666677777777777777777777777
---X- 133444555555555555555556666677777777777777777777777
X---- 133444555555555555555556666677777777777777777777777
----X 133444555555555555555556666677777777777777777777777
----- 133444555555555555555556666677777777777777777777777

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(3)a Line Value Tabulations (W8)

```
XXXXX 1111111111111111111111111111111111111111111111111111111111
XXXX- 2222222222222222222222222222222222222222222222222222222222
-XXXX 6666666666666666666666666666666666777777777777777777777777
XXX-X 1111111111111111111111111111111111111111111111111111111111
X-XXX 2222222222222222222222222222222222222222222222222222222222
XX-XX 0000011111111111111111111111111111111111111111111111111111
-XXX- 4444444444444444466666666666666677777777777777777777777777
XXX-- 5555555555555555555555555555555666666666666666666666666666
--XXX 5555555555555555555555555555555555555555555555555555555555
X-XX- 2222222222222222233333333333333444445555555555555555555555
-XX-X 0000000111111111111111111111122222222222222222222222222222
XX-X- 0000000000000000000000000000000000000000000000000000000000
-X-XX 5555555555555555555555555555555555555555555555555555555555
XX--X 2222222222222222222222222222222222222222222222222222222222
X--XX 0000000000000000000000000000000000000000000000000000000000
X-X-X 0000000000000000000000000000000000000000000000000000000000
-XX-- 4444444444444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444444444444
XX--- 3333333333333333333333333333333333333333333333333333333333
---XX 3333333333333333333333333333333333333333333333333333333333
-X-X- 2222222222222222222222222222222222222222222222222222222222
X-X-- 0000000000000000000000000000000000000000000000001111111111
--X-X 1111111111111111111111111111111111111111111111111111111111
X--X- 0000000000000000000000000000000000000000000000000000000000
-X--X 0000000000000000000000000000000000000000000000000000000000
X---X 0000000000000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000000000000

      0000000000000000000000000000000000000000000000000000000001
      5555555556666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012345678901234567890
```

```
XXXXX 6666666666666666666666666666666677777777777777777777777777
XXXX- 5555555555555555555555555555555555666666666666666666666666
-XXXX 1111111111111111111111111111111111111111111111111111111111
XXX-X 6666666666666666666666666666666677777777777777777777777777
X-XXX 5555555555555555555555555555555555666666666666666666666666
XX-XX 7777766666666666666666666666666677777777777777777777777777
-XXX- 3333333333333333311111111111111111111111111111111111111111
XXX-- 2222222222222222222222222222222122222222222222222222222222
--XXX 2222222222222222222222222222222233333333333333333333333333
X-XX- 5555555555555555554444444444444344433333333333333333333333
-XX-X 7777776666666666666666666666655566666666666666666666666666
XX-X- 7777777777777777777777777777777778888888888888888888888888
-X-XX 2222222222222222222222222222222233333333333333333333333333
XX--X 5555555555555555555555555555555566666666666666666666666666
X--XX 7777777777777777777777777777777778888888888888888888888888
X-X-X 7777777777777777777777777777777778888888888888888888888888
-XX-- 3333333333333333333333333333333344444444444444444444444444
--XX- 3333333333333333333333333333333344444444444444444444444444
XX--- 4444444444444444444444444444444555555555555555555555555555
---XX 4444444444444444444444444444444555555555555555555555555555
-X-X- 5555555555555555555555555555555566666666666666666666666666
X-X-- 7777777777777777777777777777777788888888887777777777777777
--X-X 6666666666666666666666666666666677777777777777777777777777
X--X- 7777777777777777777777777777777788888888888888888888888888
-X--X 7777777777777777777777777777777788888888888888888888888888
X---X 7777777777777777777777777777777788888888888888888888888888
--X-- 7777777777777777777777777777777788888888888888888888888888
-X--- 7777777777777777777777777777777788888888888888888888888888
---X- 7777777777777777777777777777777788888888888888888888888888
X---- 7777777777777777777777777777777788888888888888888888888888
----X 7777777777777777777777777777777788888888888888888888888888
----- 7777777777777777777777777777777788888888888888888888888888

      0000000000000000000000000000000000000000000000000000000001
      5555555556666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012345678901234567890
```

Figure 4.8(3)b continued

```
XXXXX 111111111111111111112222222222222222222222222222
XXXX- 222222222222222224477777777888888888888888888888
-XXXX 777777777777777777777777778888888888888888888888
XXX-X 111111111111111113333333333333333333333333333333
X-XXX 222222222222222222233333333333333333333333333333
XX-XX 111111111111111111111111111111111111111111111111
-XXX- 777777777777777777777777777777777777777777777777
XXX-- 666666666666666666666666666666666666666666666666
--XXX 555555555555555555555555555555555555555555555555
X-XX- 566666666666666666666666666666666666666666666666
-XX-X 222222222222222222233333333333333333333333333333
XX-X- 000001111111111112222222222222222222222222222222
-X-XX 555555555555555555555555555555555555555555555555
XX--X 223333333333333333333333333333333333333333333333
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 444444444444444444444444444444444444444444444444
--XX- 444444444444444444444444444444444444444444444444
XX--- 333333333333333333333333333333333333333333333333
---XX 333333333333333333333333333333333333333333333333
-X-X- 222222222222222222222222222222222222222222222222
X-X-- 111111111111111111111111111111111111111111111111
--X-X 111111111111111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      111111111111111111111111111111111111111111111111
      000000000111111111122222222223333333333444444444 5
      123456789012345678901234567890123456789012345678 90
```

```
XXXXX 777777777777777777776666667777777777777777777777
XXXX- 666666666666666664411111111111111111111111111111
-XXXX 111111111111111111111111111111111111111111111111
XXX-X 777777777777777775555555555666666666666666666666
X-XXX 666666666666666666655555555666666666666666666666
XX-XX 777777777777777777777777778888888888888888888888
-XXX- 111111111111111111111111111122222222222222222222
XXX-- 222222222222222222222222223333333333333333333333
--XXX 333333333333333333333333333344444444444444444444
X-XX- 322222222222222222222222223333333333333333333333
-XX-X 666666666666666666555555555666666666666666666666
XX-X- 888887777777777776666666666677777777777777777777
-X-XX 333333333333333333333333333344444444444444444444
XX--X 665555555555555555555555555666666666666666666666
X--XX 888888888888888888888888888999999999999999999999
X-X-X 888888888888888888888888888999999999999999999999
-XX-- 444444444444444444444444444455555555555555555555
--XX- 444444444444444444444444444455555555555555555555
XX--- 555555555555555555555555555566666666666666666666
---XX 555555555555555555555555555566666666666666666666
-X-X- 666666666666666666666666666677777777777777777777
X-X-- 777777777777777777777777777788888888888888888888
--X-X 777777777777777777777777777788888888888888888888
X--X- 888888888888888888888888888899999999999999999999
-X--X 888888888888888888888888888899999999999999999999
X---X 888888888888888888888888888899999999999999999999
--X-- 888888888888888888888888888899999999999999999999
-X--- 888888888888888888888888888899999999999999999999
---X- 888888888888888888888888888899999999999999999999
X---- 888888888888888888888888888899999999999999999999
----X 888888888888888888888888888899999999999999999999
----- 888888888888888888888888888899999999999999999999

      111111111111111111111111111111111111111111111111
      000000000111111111122222222223333333333444444444 5
      123456789012345678901234567890123456789012345678 90
```

Figure 4.8(3)c continued

```
XXXXX 2222223333333333333333333333333333333333333333333333
XXXX- 8888888888888888888888888888888888888888888888999999
-XXXX 8888888888888888888888888888888888888888888888999999
XXX-X 3333333333333333333333333333333333333333333333333333
X-XXX 3333333333333333333333333333333333333333333333333333
XX-XX 1111111111111111111111111111111111111111111111111111
-XXX- 7777777777777777777777777777777777777777777778888888
XXX-- 6666666666666666666666666666666666666666666666666666
--XXX 5555555555555555555555555555555556666666666667777777
X-XX- 6666666666666666666666666666666666666666666666666666
-XX-X 3333333444444444444444444444445555555555555555555555
XX-X- 2222222222222222222222222222222222222222222223333333
-X-XX 5555555555555555555555555555555555555555555555555555
XX--X 3333333333333333333333333333333333333333333355555555
X--XX 0000000000000000000000000000000000000000000000000000
X-X-X 0000000000000000000000000000000000000000000000000000
-XX-- 4444444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444444
XX--- 3333333333333333333333333333333333333333333333333333
---XX 3333333333333333333333333333333333333333333333333333
-X-X- 2222222222222222222222222222222222222222222222222222
X-X-- 1111111111111111111111111111111111111111111111111111
--X-X 1111111111111111111111111111111111111111111111111111
X--X- 0000000000000000000000000000000000000000000000000000
-X--X 0000000000000000000000000000000000000000000000000000
X---X 0000000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000000

      1111111111111111111111111111111111111111111111111112
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012
```

```
XXXXX 7777776666666666666666666666666666666666666666777777
XXXX- 1111111111111111111111111111111111111111111111111111
-XXXX 1111111111111111111111111111111111111111111111111111
XXX-X 6666666666666666666666666666666666666666666666777777
X-XXX 6666666666666666666666666666666666666666666666777777
XX-XX 8888888888888888888888888888888888888888888888999999
-XXX- 2222222222222222222222222222222222222222222221222222
XXX-- 3333333333333333333333333333333333333333333333444444
--XXX 4444444444444444444444444444443333333333333222333333
X-XX- 3333333333333333333333333333333333333333333333444444
-XX-X 6666666555555555555555555555444444444444444555555
XX-X- 7777777777777777777777777777777777777777777777777777
-X-XX 4444444444444444444444444444444444444444444555555
XX--X 6666666666666666666666666666666666666666666445555555
X--XX 9999999999999999999999999999999999999999999AAAAAA
X-X-X 9999999999999999999999999999999999999999999AAAAAA
-XX-- 5555555555555555555555555555555555555555555666666
--XX- 5555555555555555555555555555555555555555555666666
XX--- 6666666666666666666666666666666666666666666777777
---XX 6666666666666666666666666666666666666666666777777
-X-X- 7777777777777777777777777777777777777777777888888
X-X-- 8888888888888888888888888888888888888888888999999
--X-X 8888888888888888888888888888888888888888888999999
X--X- 9999999999999999999999999999999999999999999AAAAAA
-X--X 9999999999999999999999999999999999999999999AAAAAA
X---X 9999999999999999999999999999999999999999999AAAAAA
--X-- 9999999999999999999999999999999999999999999AAAAAA
-X--- 9999999999999999999999999999999999999999999AAAAAA
---X- 9999999999999999999999999999999999999999999AAAAAA
X---- 9999999999999999999999999999999999999999999AAAAAA
----X 9999999999999999999999999999999999999999999AAAAAA
----- 9999999999999999999999999999999999999999999AAAAAA

      1111111111111111111111111111111111111111111111111112
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012
```

Figure 4.8(3)d continued

```
XXXXX 000000000000000000000000000000001111111111111111
XXXX- 000000000000000000000000111111111111111111111111
-XXXX 000000000000000000000000000000000000000000000000
XXX-X 000000000000000000000000111111111111111111111111
X-XXX 000000111111111111111111111111222222222222222222
XX-XX 000000000000000000000000000000000000000000000000
-XXX- 000001111111111111111111111111111111111111111111
XXX-- 000000000000000000000000000000000000000000000000
--XXX 000000444444444444444444444444555555555555666666
X-XX- 000111222222222222222222222222222222222255555555
-XX-X 000001111111111111111111111111111111111111111111
XX-X- 000000000000000000000000000000000000000000000000
-X-XX 000000000000000000000000000000000000000000000000
XX--X 000000000000000000000000000000000000000000000000
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 011111113333344444444444444444444444444444444444
--XX- 033333333333344444444444444444444444444444444444
XX--- 022222222222233333333333333333333333333333333333
---XX 000000111111111111333333333333333333333333333333
-X-X- 011111111111122222222222222222222222222222222222
X-X-- 000000000000000000000000000000000000000000000000
--X-X 000000000001111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 144444555555555555555555555555666655555556666666666
XXXX- 144444555555555555555555555444455555555556666666666
-XXXX 144444555555555555555555555555555666666666677777777777
XXX-X 144444555555555555555555555444455555555556666666666
X-XXX 144444444444444444444444444444544444444445555555555
XX-XX 144444555555555555555555555555555666666666677777777777
-XXX- 144443444444444444444444444444455555555556666666666
XXX-- 144444555555555555555555555555555666666666677777777777
--XXX 144444111111111111111111111111111111111111111111111
X-XX- 144333333333333333333333333333344444444441222222222
-XX-X 144443444444444444444444444444455555555556666666666
XX-X- 144444555555555555555555555555555666666666677777777777
-X-XX 144444555555555555555555555555555666666666677777777777
XX--X 144444555555555555555555555555555666666666677777777777
X--XX 144444555555555555555555555555555666666666677777777777
X-X-X 144444555555555555555555555555555666666666677777777777
-XX-- 133333442222211111111111111111112222222222333333333333
--XX- 111111222222211111111111111111112222222222333333333333
XX--- 122222333333322222222222222222233333333333444444444444
---XX 144444444444444444222222222222223333333333444444444444
-X-X- 133333444444433333333333333333334444444444555555555555
X-X-- 144444555555555555555555555555555666666666677777777777
--X-X 144444555555444444444444444444555555555556666666666
X--X- 144444555555555555555555555555555666666666677777777777
-X--X 144444555555555555555555555555555666666666677777777777
X---X 144444555555555555555555555555555666666666677777777777
--X-- 144444555555555555555555555555555666666666677777777777
-X--- 144444555555555555555555555555555666666666677777777777
---X- 144444555555555555555555555555555666666666677777777777
X---- 144444555555555555555555555555555666666666677777777777
----X 144444555555555555555555555555555666666666677777777777
----- 144444555555555555555555555555555666666666677777777777

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(4)a Line Value Tabulations (WC)

```
XXXXX 1111111111111111111111111111111111111111111111111111
XXXX- 1111111111111111111111111111111111111111111111111111
-XXXX 0000000000000000000000000000000000000000000000000000
XXX-X 1111111111111111111111111111111111111111111111111111
X-XXX 2222222222222222222222222222222222222222222222222222
XX-XX 0000000000000000000000000000000000000000000000000000
-XXX- 1111111111111111111111111111111111111111111111111111
XXX-- 0000000000000000000000000000000000000000000000000000
--XXX 6666666666666666666666666666666666666666666666666666
X-XX- 5555555555555555555555555555555555555555555555555555
-XX-X 1111111111111111111111111111111111111111111111111111
XX-X- 0000000000000000000000000000000000000000000000000000
-X-XX 0000000000000000000000000000000000000000000000000000
XX--X 0000000000000000000000000000000000000000000000000000
X--XX 0000000000000000000000000000000000000000000000000000
X-X-X 0000000000000000000000000000000000000000000000000000
-XX-- 4444444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444444
XX--- 3333333333333333333333333333333333333333333333333333
---XX 3333333333333333333333333333333333333333333333333333
-X-X- 2222222222222222222222222222222222222222222222222222
X-X-- 0000000000000000000000000000000000000000000000000000
--X-X 1111111111111111111111111111111111111111111111111111
X--X- 0000000000000000000000000000000000000000000000000000
-X--X 0000000000000000000000000000000000000000000000000000
X---X 0000000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000000

      0000000000000000000000000000000000000000000000000001
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012
```

```
XXXXX 6666666666666666666666666666666666666666666666666666
XXXX- 6666666666666666666666666666666666666666666666666666
-XXXX 7777777777777777777777777777777777777777777777777777
XXX-X 6666666666666666666666666666666666666666666666666666
X-XXX 5555555555555555555555555555555555555555555555555555
XX-XX 7777777777777777777777777777777777777777777777777777
-XXX- 6666666666666666666666666666666666666666666666666666
XXX-- 7777777777777777777777777777777777777777777777777777
--XXX 1111111111111111111111111111111111111111111111111111
X-XX- 2222222222222222222222222222222222222222222222222222
-XX-X 6666666666666666666666666666666666666666666666666666
XX-X- 7777777777777777777777777777777777777777777777777777
-X-XX 7777777777777777777777777777777777777777777777777777
XX--X 7777777777777777777777777777777777777777777777777777
X--XX 7777777777777777777777777777777777777777777777777777
X-X-X 7777777777777777777777777777777777777777777777777777
-XX-- 3333333333333333333333333333333333333333333333333333
--XX- 3333333333333333333333333333333333333333333333333333
XX--- 4444444444444444444444444444444444444444444444444444
---XX 4444444444444444444444444444444444444444444444444444
-X-X- 5555555555555555555555555555555555555555555555555555
X-X-- 7777777777777777777777777777777777777777777777777777
--X-X 6666666666666666666666666666666666666666666666666666
X--X- 7777777777777777777777777777777777777777777777777777
-X--X 7777777777777777777777777777777777777777777777777777
X---X 7777777777777777777777777777777777777777777777777777
--X-- 7777777777777777777777777777777777777777777777777777
-X--- 7777777777777777777777777777777777777777777777777777
---X- 7777777777777777777777777777777777777777777777777777
X---- 7777777777777777777777777777777777777777777777777777
----X 7777777777777777777777777777777777777777777777777777
----- 7777777777777777777777777777777777777777777777777777

      0000000000000000000000000000000000000000000000000001
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012
```

Figure 4.8(4)b continued

```
XXXXX 111111111111111111111111111111111111111111111111111
XXXX- 111111111111111111111111111111111111111114444444444444
-XXXX 000000000000000000000000000000000000000000000000000
XXX-X 111111111111111111111111111111111111111111111111111
X-XXX 222222222224444444444444444445555555555555555555555
XX-XX 000000000000000000000000000000000000000000000000000
-XXX- 111111111111122222222222222222222222222222222222222
XXX-- 000011111111111111111111111111111111111111111111111
--XXX 666666666666666666666666666666666666666666666666666
X-XX- 555555555555555555555555555555555555555555555555555
-XX-X 111111111111111111111111111111111111111111122222222
XX-X- 000000000000000000000000000000000000011111111111111
-X-XX 000000001111222222222222222222222222222222222222222
XX--X 000000000000000000000000000000000000000000000000000
X--XX 000000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000000
-XX-- 444444444444444444444444444444444444444444444444444
--XX- 444444444444444444444444444444444444444444444444444
XX--- 333333333333333333333333333333333333333333333333333
---XX 333333333333333333333333333333333333333333333333333
-X-X- 222222222222222222222222222222222222222222222222222
X-X-- 000000000000000000000000000000000000000000000000000
--X-X 111111111111111111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000000

      111111111111111111111111111111111111111111111111111
      000000000111111111122222222223333333333344444444445
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 666666666666666666666666666666666666666666666666666
XXXX- 666666666666666666666666666666666666663333333333333
-XXXX 777777777777777777777777777777777777777777777777777
XXX-X 666666666666666666666666666666666666666666666666666
X-XXX 555555555553333333333333333332222222222222222222222
XX-XX 777777777777777777777777777777777777777777777777777
-XXX- 666666666666655555555555555555555555555555555555555
XXX-- 777766666666666666666666666666666666666666666666666
--XXX 111111111111111111111111111111111111111111111111111
X-XX- 222222222222222222222222222222222222222222222222222
-XX-X 666666666666666666666666666666666666666665555555555
XX-X- 777777777777777777777777777777777777776666666666666
-X-XX 777777776666555555555555555555555555555555555555555
XX--X 777777777777777777777777777777777777777777777777777
X--XX 777777777777777777777777777777777777777777777777777
X-X-X 777777777777777777777777777777777777777777777777777
-XX-- 333333333333333333333333333333333333333333333333333
--XX- 333333333333333333333333333333333333333333333333333
XX--- 444444444444444444444444444444444444444444444444444
---XX 444444444444444444444444444444444444444444444444444
-X-X- 555555555555555555555555555555555555555555555555555
X-X-- 777777777777777777777777777777777777777777777777777
--X-X 666666666666666666666666666666666666666666666666666
X--X- 777777777777777777777777777777777777777777777777777
-X--X 777777777777777777777777777777777777777777777777777
X---X 777777777777777777777777777777777777777777777777777
--X-- 777777777777777777777777777777777777777777777777777
-X--- 777777777777777777777777777777777777777777777777777
---X- 777777777777777777777777777777777777777777777777777
X---- 777777777777777777777777777777777777777777777777777
----X 777777777777777777777777777777777777777777777777777
----- 777777777777777777777777777777777777777777777777777

      111111111111111111111111111111111111111111111111111
      000000000111111111122222222223333333333344444444445
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(4)c continued

```
XXXXX 111222222222222222222222222222222222222222222222
XXXX- 444444444444444444445555577777777777777777777777
-XXXX 000000000111111111111111111111111111111111111111
XXX-X 111111111111222222222222222222222222222233333333
X-XXX 555555555555555555555555555555555555555555555555
XX-XX 000000000000000000000000000000000000000000000000
-XXX- 222222222222222222222222222222222222222222222222
XXX-- 111111111111111111111111111111111111111111111111
--XXX 666666666666666666666666666666666666666666666666
X-XX- 555555555555555555555555555555555555555555555555
-XX-X 222222222222222222222222222222222222333333335555
XX-X- 111111111111111111111111111111222222222222222222
-X-XX 222222222222222222222222222222222222222222344444
XX--X 000000000000000000000000000000000000000000000000
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 444444444444444444444444444444444444444444444444
--XX- 444444444444444444444444444444444444444444444444
XX--- 333333333333333333333333333333333333333333333333
---XX 333333333333333333333333333333333333333333333333
-X-X- 222222222222222222222222222222222222222222222222
X-X-- 000000000000000000000000000011111111111111111111
--X-X 111111111111111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      111111111111111111111111111111111111111111111112
      555555555566666666667777777777888888888899999999990
      123456789012345678901234567890123456789012345678901234567890
```

```
XXXXX 666555555555555555555555566666666666666666666666
XXXX- 333333333333333333332222211111111111111111111111
-XXXX 777777777666666666666666677777777777777777777777
XXX-X 666666666666655555555555555666666666666666655555555
X-XXX 222222222222222222222222233333333333333333333333
XX-XX 777777777777777777777777788888888888888888888888
-XXX- 555555555555555555555555566666666666666666666666
XXX-- 666666666666666666666666677777777777777777777777
--XXX 111111111111111111111111122222222222222222222222
X-XX- 555555555555555555555555533333333333333333333333
-XX-X 555555555555555555555555566666665555555533333333
XX-X- 666666666666666666666666677666666666666666666666
-X-XX 555555555555555555555555566666666666666665444444
XX--X 777777777777777777777777788888888888888888888888
X--XX 777777777777777777777777788888888888888888888888
X-X-X 777777777777777777777777788888888888888888888888
-XX-- 333333333333333333333333344444444444444444444444
--XX- 333333333333333333333333344444444444444444444444
XX--- 444444444444444444444444455555555555555555555555
---XX 444444444444444444444444455555555555555555555555
-X-X- 555555555555555555555555566666666666666666666666
X-X-- 777777777777777777777777788777777777777777777777
--X-X 666666666666666666666666677777777777777777777777
X--X- 777777777777777777777777788888888888888888888888
-X--X 777777777777777777777777788888888888888888888888
X---X 777777777777777777777777788888888888888888888888
--X-- 777777777777777777777777788888888888888888888888
-X--- 777777777777777777777777788888888888888888888888
---X- 777777777777777777777777788888888888888888888888
X---- 777777777777777777777777788888888888888888888888
----X 777777777777777777777777788888888888888888888888
----- 777777777777777777777777788888888888888888888888

      111111111111111111111111111111111111111111111112
      555555555566666666667777777777888888888899999999990
      123456789012345678901234567890123456789012345678901234567890
```

Figure 4.8(4)d continued

```
XXXXX 000000000000000000000000000000000000000000000000
XXXX- 000000000000111333333333333333333333333333333333
-XXXX 000000000000111111111111111111111111111111111111
XXX-X 000000000000000000000000000000000000000011122222
X-XXX 000000000000234444444444444444444444444445555555
XX-XX 000000000000000000000000000000000000000000000000
-XXX- 000000000134444444444444444444444444555555555555
XXX-- 000000000011111111111111122222334444444456666666
--XXX 000000000000000000001111111111111111111111111111
X-XX- 000011112233333333333444444444444444555555555555
-XX-X 000000000000000000001111111111111111111111111111
XX-X- 000000000000000011111111111111111111111111111111
-X-XX 000000000011122222222333333333333334444444444444
XX--X 000000000000000000000000000000000000000000000000
X--XX 000000011111111111111111111111111111111111111111
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 033333333333333333333333333333333333333333333333
--XX- 033333333333333333333333333333333333333333333333
XX--- 012222222222222222222222222222222222222222222222
---XX 022222222222222222222222222222222222222222222222
-X-X- 011111111111111111111111111111111111111111111111
X-X-- 000000000000000000000000000000000000000000000001
--X-X 000000000000000000000000000000000000000000000001
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 1444444444455555555555555555555555555666666666677777777
XXXX- 1444444444455444222222222222222222223333333333444444444
-XXXX 1444444444435444444444444444444444444555555555566666666
XXX-X 1444444444455555555555555555555555555566666555555555555
X-XXX 1444444444455321111111111111111111111222222111222222222
XX-XX 1444444444455555555555555555555555555566666666667777777
-XXX- 1444444443111111111111111111111111111111111111222222222
XXX-- 1444444444444444444444444444333333221222222221111111111
--XXX 1444444444455555555555444444444444445555555555566666666
X-XX- 1444333322222222222221111111111111122111111112222222222
-XX-X 1444444444455555555555444444444444445555555555566666666
XX-X- 1444444444455555444444444444444444445555555555566666666
-X-XX 1444444444444333333332222222222222333222222223333333333
XX--X 1444444444455555555555555555555555555566666666667777777
X--XX 1444444333344444444444444444444444445555555555566666666
X-X-X 1444444444455555555555555555555555555566666666667777777
-XX-- 1111111111122222222222222222222222223333333333444444444
--XX- 1111111111122222222222222222222222223333333333444444444
XX--- 1322222222233333333333333333333333333444444444455555555
---XX 1222222222233333333333333333333333333444444444455555555
-X-X- 1333333333344444444444444444444444445555555555566666666
X-X-- 1444444444455555555555555555555555555566666666667777776
--X-X 1444444444455555555555555555555555555566666666667777776
X--X- 1444444444455555555555555555555555555566666666667777777
-X--X 1444444444455555555555555555555555555566666666667777777
X---X 1444444444455555555555555555555555555566666666667777777
--X-- 1444444444455555555555555555555555555566666666667777777
-X--- 1444444444455555555555555555555555555566666666667777777
---X- 1444444444455555555555555555555555555566666666667777777
X---- 1444444444455555555555555555555555555566666666667777777
----X 1444444444455555555555555555555555555566666666667777777
----- 1444444444455555555555555555555555555566666666667777777

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(5)a Line Value Tabulations (W4)

```
XXXXX 0000000000000000000000000000000000000000000000000
XXXX- 3333333333333334444444444444444444555555555555555
-XXXX 1111111111111111111111111111111111111111111111111
XXX-X 2222222222222222222222222222222222222222222222222
X-XXX 5555555555555555555555555556666666666666666666666
XX-XX 0000000000000000000000000000000000000000000000000
-XXX- 5555555555555555555555555555555555555555555555555
XXX-- 6666666666666666666666666666666666666666666666666
--XXX 1111111111111111111111112222222222222222222222222
X-XX- 5555555555555555555555555555555555556666666666666
-XX-X 1111222222222222222222222222222222222222222223444
XX-X- 2222222222222222222222222222222222222222222222222
-X-XX 4444444444444445555555555555555555555555555555555
XX--X 0000000000000000000000000000000000000000000000000
X--XX 1111111111111111111111111111111111111111111111111
X-X-X 0000000000000000000000000000000000000000000000000
-XX-- 3333333333344444444444444444444444444444444444444
--XX- 3333333333344444444444444444444444444444444444444
XX--- 2222222222233333333333333333333333333333333333333
---XX 2222222222233333333333333333333333333333333333333
-X-X- 2222222222222222222222222222222222222222222222222
X-X-- 1111111111111111111111111111111111111111111111111
--X-X 1111111111111111111111111111111111111111111111111
X--X- 0000000000000000000000000000000000000000000000000
-X--X 0000000000000000000000000000000000000000000000000
X---X 0000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000

      0000000000000000000000000000000000000000000000001
      5555555553666666666677777777778888888888999999999 0
      1234567890123456789012345678901234567890123456789 0
```

```
XXXXX 7777777777777777777777777777777777777777777777777
XXXX- 4444444444444443333333333333333333322222222222222
-XXXX 6666666666666666666666666666666666666666666666666
XXX-X 5555555555555555555555555555555555555555555555555
X-XXX 2222222222222222222222221111111111111111111111111
XX-XX 7777777777777777777777777777777777777777777777777
-XXX- 2222222222222222222222222222222222222222222222222
XXX-- 1111111111111111111111111111111111111111111111111
--XXX 6666666666666666666666555555555555555555555555555
X-XX- 2222222222222222222222222222222222221111111111111
-XX-X 6666555555555555555555555555555555555555555554333
XX-X- 5555555555555555555555555555555555555555555555555
-X-XX 3333333333333332222222222222222222222222222222222
XX--X 7777777777777777777777777777777777777777777777777
X--XX 6666666666666666666666666666666666666666666666666
X-X-X 7777777777777777777777777777777777777777777777777
-XX-- 4444444444433333333333333333333333333333333333333
--XX- 4444444444433333333333333333333333333333333333333
XX--- 5555555555544444444444444444444444444444444444444
---XX 5555555555544444444444444444444444444444444444444
-X-X- 5555555555555555555555555555555555555555555555555
X-X-- 6666666666666666666666666666666666666666666666666
--X-X 6666666666666666666666666666666666666666666666666
X--X- 7777777777777777777777777777777777777777777777777
-X--X 7777777777777777777777777777777777777777777777777
X---X 7777777777777777777777777777777777777777777777777
--X-- 7777777777777777777777777777777777777777777777777
-X--- 7777777777777777777777777777777777777777777777777
---X- 7777777777777777777777777777777777777777777777777
X---- 7777777777777777777777777777777777777777777777777
----X 7777777777777777777777777777777777777777777777777
----- 7777777777777777777777777777777777777777777777777

      0000000000000000000000000000000000000000000000001
      5555555553666666666677777777778888888888999999999 0
      1234567890123456789012345678901234567890123456789 0
```

Figure 4.8(5)b continued

```
XXXXX 000000000000000000000000000000000000000000000000
XXXX- 555555555555555555555555555555555555555556666677 77
-XXXX 111111111111111111111111111111111111111111111111
XXX-X 222222333333333333333333333333333333333333333333
X-XXX 666666666677777777777777777777777777777777777777
XX-XX 000000000000000000000000000000000000000000000000
-XXX- 555555555555555555555555555555555555555555567777
XXX-- 666666666666666666666666666666666666666666666666
--XXX 222222222222222222222222222234444444444455555555
X-XX- 666666666666666666666666666666666666666666666666
-XX-X 444455555555555555555555555555555555555555555555
XX-X- 222222222223333333333333333334444444444444444445
-X-XX 555555555555555555555555555555555555555555555555
XX--X 000000000000000000000000000000000000000000000000
X--XX 111111111111111111111111111111111111111111111111
-X-X-X 000000000000000000000000000000000000000000000000
-XX-- 444444444444444444444444444444444444444444444444
--XX- 444444444444444444444444444444444444444444444444
XX--- 333333333333333333333333333333333333333333333333
---XX 333333333333333333333333333333333333333333333333
-X-X- 222222222222222222222222222222222222222222222222
X-X-- 111111111111111111111111111111111111111111111111
--X-X 111111111111111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      111111111111111111111111111111111111111111111111
      000000000111111111122222222223333333333444444444 5
      123456789012345678901234567890123456789012345678 90
```

```
XXXXX 777777777788888888888888888888888888888888888888
XXXX- 222222222233333333333333333333333333333222222111 1
-XXXX 666666666677777777777777777777777777777777777777
XXX-X 555555444455555555555555555555555555555555555555
X-XXX 111111111111111111111111111111111111111111111111
XX-XX 777777777788888888888888888888888888888888888888
-XXX- 222222222233333333333333333333333333333321111111
XXX-- 111111111122222222222222222222222222222222222222
--XXX 555555555566666666666666666665444444444443333333
X-XX- 111111111122222222222222222222222222222222222222
-XX-X 333322222233333333333333333333333333333333333333
XX-X- 555555555565555555555555555555444444444444444443
-X-XX 222222222233333333333333333333333333333333333333
XX--X 777777777788888888888888888888888888888888888888
X--XX 666666666677777777777777777777777777777777777777
X-X-X 777777777788888888888888888888888888888888888888
-XX-- 333333333344444444444444444444444444444444444444
--XX- 333333333344444444444444444444444444444444444444
XX--- 444444444455555555555555555555555555555555555555
---XX 444444444455555555555555555555555555555555555555
-X-X- 555555555566666666666666666666666666666666666666
X-X-- 666666666677777777777777777777777777777777777777
--X-X 666666666677777777777777777777777777777777777777
X--X- 777777777788888888888888888888888888888888888888
-X--X 777777777788888888888888888888888888888888888888
X---X 777777777788888888888888888888888888888888888888
--X-- 777777777788888888888888888888888888888888888888
-X--- 777777777788888888888888888888888888888888888888
---X- 777777777788888888888888888888888888888888888888
X---- 777777777788888888888888888888888888888888888888
----X 777777777788888888888888888888888888888888888888
----- 777777777788888888888888888888888888888888888888

      111111111111111111111111111111111111111111111111
      000000000111111111122222222223333333333444444444 5
      123456789012345678901234567890123456789012345678 90
```

Figure 4.8(5)c continued

```
XXXXX 0000000000000000000000000000000000000000000000000000
XXXX- 7777777777777777777777777777777777777777777777777777
-XXXX 1111111111111111111111111111111111111112222222222222
XXX-X 3333333333333333333333333333333333444444444444444444
X-XXX 7777777777777777777777777777777777777777777777777777
XX-XX 0000000000000000000000000000000000000000000000000000
-XXX- 7777777777777777777777777777777777777777777777777777
XXX-- 6666666666666666666666666666666666666666666666666666
--XXX 5555555555555555555555555555555555555555555555555555
X-XX- 6666666666666666666666666666666666666666666666666666
-XX-X 5555555555555555555555555555555555566666666666666666
XX-X- 5555555555555555555555555555555555555555555555555555
-X-XX 5555555555555555555555555555555555555555555555555555
XX--X 0000000000000000000000000000000000000000000000000000
X--XX 1111111111111111111111111111111111111111111111111111
X-X-X 0000000000000000000000000000000000000000000000000000
-XX-- 4444444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444444
XX--- 3333333333333333333333333333333333333333333333333333
---XX 3333333333333333333333333333333333333333333333333333
-X-X- 2222222222222222222222222222222222222222222222222222
X-X-- 1111111111111111111111111111111111111111111111111111
--X-X 1111111111111111111111111111111111111111111111111111
X--X- 0000000000000000000000000000000000000000000000000000
-X--X 0000000000000000000000000000000000000000000000000000
X---X 0000000000000000000000000000000000000000000000000000
--X-- 0000000000000000000000000000000000000000000000000000
-X--- 0000000000000000000000000000000000000000000000000000
---X- 0000000000000000000000000000000000000000000000000000
X---- 0000000000000000000000000000000000000000000000000000
----X 0000000000000000000000000000000000000000000000000000
----- 0000000000000000000000000000000000000000000000000000

      1111111111111111111111111111111111111111111111111112
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012
```

```
XXXXX 8888888888888888888888888888888888888888888888888888
XXXX- 1111111111111111111111111111111111111111111111111111
-XXXX 7777777777777777777777777777777777666666666666666666
XXX-X 5555555555555555555555555555555555444444444444444444
X-XXX 1111111111111111111111111111111111111111111111111111
XX-XX 8888888888888888888888888888888888888888888888888888
-XXX- 1111111111111111111111111111111111111111111111111111
XXX-- 2222222222222222222222222222222222222222222222222222
--XXX 3333333333333333333333333333333333333333333333333333
X-XX- 2222222222222222222222222222222222222222222222222222
-XX-X 3333333333333333333333333333333333222222222222222222
XX-X- 3333333333333333333333333333333333333333333333333333
-X-XX 3333333333333333333333333333333333333333333333333333
XX--X 8888888888888888888888888888888888888888888888888888
X--XX 7777777777777777777777777777777777777777777777777777
X-X-X 8888888888888888888888888888888888888888888888888888
-XX-- 4444444444444444444444444444444444444444444444444444
--XX- 4444444444444444444444444444444444444444444444444444
XX--- 5555555555555555555555555555555555555555555555555555
---XX 5555555555555555555555555555555555555555555555555555
-X-X- 6666666666666666666666666666666666666666666666666666
X-X-- 7777777777777777777777777777777777777777777777777777
--X-X 7777777777777777777777777777777777777777777777777777
X--X- 8888888888888888888888888888888888888888888888888888
-X--X 8888888888888888888888888888888888888888888888888888
X---X 8888888888888888888888888888888888888888888888888888
--X-- 8888888888888888888888888888888888888888888888888888
-X--- 8888888888888888888888888888888888888888888888888888
---X- 8888888888888888888888888888888888888888888888888888
X---- 8888888888888888888888888888888888888888888888888888
----X 8888888888888888888888888888888888888888888888888888
----- 8888888888888888888888888888888888888888888888888888

      1111111111111111111111111111111111111111111111111112
      5555555555666666666677777777778888888888999999999990
      1234567890123456789012345678901234567890123456789012
```

Figure 4.8(5)d continued

```
XXXXX 000000000000000000000000000000000000000000000000000
XXXX- 000000000000000111111111111111111111111111111222222
-XXXX 000000000000000000000000000011111111111222222222222
XXX-X 000000000000000000000000000000000000000000000000000
X-XXX 000000000000000000000000000000000000000000000000000
XX-XX 000000033333333555555555555555555555555555555555555
-XXX- 000000011111111111111111111111111111111222222222222
XXX-- 000000000000000000000000000000000011111111111111111
--XXX 000000000000000000000000000000000000000000000000000
X-XX- 000000023333333444444444445555555555555555555555555
-XX-X 000000011111111111111333333444445555555555555555555
XX-X- 000000000000111111111111111111111111111111111111111
-X-XX 000000000111111111111111112233333333333333333333333
XX--X 000000000000000000000000000000011111111111111111111
X--XX 000000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000000
-XX-- 033333333333333333333344444444444444444444444444444
--XX- 033333333333333333333344444444444444444444444444444
XX--- 012222222222222222222333333333333333333333333333333
---XX 022222222222222222222333333333333333333333333333333
-X-X- 011111111111111111112222222222222222222222222222222
X-X-- 000000000000000000000000000000000000000000000000000
--X-X 000000000000011111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000000

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 144444444444446666666666666666666666666666666666666
XXXX- 144444444444446555555555555555555555555555554444444
-XXXX 144444444444446666666666666666555555555555444444444
XXX-X 144444444444446666666666666666666666666666666666666
X-XXX 144444444444446666666666666666666666666666666666666
XX-XX 144444111111111111111111111111111111111111111111111
-XXX- 144444433333333555555555555555555555555444444444444
XXX-- 144444444444446666666666666666666655555555555555555
--XXX 144444444444446666666666666666666666666666666666666
X-XX- 144444211111111222222222221111111111111111111111111
-XX-X 144444333333333555555333333222221111111111111111111
XX-X- 144444444443335555555555555555555555555555555555555
-X-XX 144444443333335555555555554433333333333333333333333
XX--X 144444444444446666666666666666666655555555555555555
X--XX 144444444444446666666666666666666666666666666666666
X-X-X 144444444444446666666666666666666666666666666666666
-XX-- 111111111111111333333322222222222222222222222222222
--XX- 111111111111111333333322222222222222222222222222222
XX--- 132222222222224444444433333333333333333333333333333
---XX 122222222222224444444433333333333333333333333333333
-X-X- 133333333333335555554444444444444444444444444444444
X-X-- 144444444444446666666666666666666666666666666666666
--X-X 144444444444435555555555555555555555555555555555555
X--X- 144444444444446666666666666666666666666666666666666
-X--X 144444444444446666666666666666666666666666666666666
X---X 144444444444446666666666666666666666666666666666666
--X-- 144444444444446666666666666666666666666666666666666
-X--- 144444444444446666666666666666666666666666666666666
---X- 144444444444446666666666666666666666666666666666666
X---- 144444444444446666666666666666666666666666666666666
----X 144444444444446666666666666666666666666666666666666
----- 144444444444446666666666666666666666666666666666666

      000000000111111111122222222223333333333444444444445
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(6)a Line Value Tabulations (WE)

```
XXXXX 000000000000000000000000000000000000000000000000
XXXX- 222222222222222222222222222222222222222222222222
-XXXX 222222222225555555555555555555555555555555555555
XXX-X 000000000000000000000000000000000000000000000011
X-XXX 000000000000000000000000000000000000000000000000
XX-XX 555555555555555555555555555555555555555555555555
-XXX- 222222222222222222222222222222222222225555555555
XXX-- 111111111111111111111111111111111111111111111122
--XXX 000000000000000000000000000000004444444444444444
X-XX- 555555555555555555555555556666666666666666666666
-XX-X 555555555555555555555555556666666666666666666666
XX-X- 111111111111111111111111111111111111111111333333
-X-XX 333333333444444444444444555555555555555555555555
XX--X 111111111111111111111111111111111111111111111111
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 444444444444444444444444444444444444444444444444
--XX- 444444444444444444444444444444444444444444444444
XX--- 333333333333333333333333333333333333333333333333
---XX 333333333333333333333333333333333333333333333333
-X-X- 222222222222222222222222222222222222222222222222
X-X-- 000000000000000000000000000000000000000000111111
--X-X 111111111111111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      000000000000000000000000000000000000000000000001
      555555555566666666667777777777888888888899999999990
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 666666666666666666666666677777777777777777777777777
XXXX- 444444444444444444444444555555555555555555555555555
-XXXX 444444444441111111111111222222222222222222222222222
XXX-X 666666666666666666666666677777777777777777777777766
X-XXX 666666666666666666666666677777777777777777777777777
XX-XX 111111111111111111111111222222222222222222222222222
-XXX- 444444444444444444444444555555555555555555222222222
XXX-- 555555555555555555555555666666666666666666666666655
--XXX 666666666666666666666666677777777777333333333333333
X-XX- 111111111111111111111111111111111111111111111111111
-XX-X 111111111111111111111111222111111111111111111111111
XX-X- 555555555555555555555555666666666666666666666444444
-X-XX 333333333222222222222222222222222222222222222222222
XX--X 555555555555555555555555666666666666666666666666666
X--XX 666666666666666666666666677777777777777777777777777
X-X-X 666666666666666666666666677777777777777777777777777
-XX-- 222222222222222222222222333333333333333333333333333
--XX- 222222222222222222222222333333333333333333333333333
XX--- 333333333333333333333333444444444444444444444444444
---XX 333333333333333333333333444444444444444444444444444
-X-X- 444444444444444444444444555555555555555555555555555
X-X-- 666666666666666666666666677777777777777777777666666
--X-X 555555555555555555555555666666666666666666666666666
X--X- 666666666666666666666666677777777777777777777777777
-X--X 666666666666666666666666677777777777777777777777777
X---X 666666666666666666666666677777777777777777777777777
--X-- 666666666666666666666666677777777777777777777777777
-X--- 666666666666666666666666677777777777777777777777777
---X- 666666666666666666666666677777777777777777777777777
X---- 666666666666666666666666677777777777777777777777777
----X 666666666666666666666666677777777777777777777777777
----- 666666666666666666666666677777777777777777777777777

      000000000000000000000000000000000000000000000000001
      555555555566666666667777777777888888888899999999990
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(6)b continued

```
XXXXX 000000000000000000000000000000000000000000000000
XXXX- 222222222222222334444445555555555555555556666666
-XXXX 555555556666666666666666666666666888888888888888
XXX-X 111111111111111111111111111111111111111111111111
X-XXX 000000000000000000000000000000000000000000000000
XX-XX 555555555555555555555555555555555555555555555555
-XXX- 555555666666666666666666666666666677777777777777
XXX-- 222222222222222222222222222222222222222222222222
--XXX 444444455555555555555555555555555555555555555555
X-XX- 666666666666666666666666666666666666666666666666
-XX-X 666666666666666666666666666666666666666666666666
XX-X- 333333333333333333333333333333333333333333333333
-X-XX 555555555555555555555555555555555555555555555555
XX--X 111111111111111111111111111111111111111111111111
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 444444444444444444444444444444444444444444444444
--XX- 444444444444444444444444444444444444444444444444
XX--- 333333333333333333333333333333333333333333333333
---XX 333333333333333333333333333333333333333333333333
-X-X- 222222222222222222222222222222222222222222222222
X-X-- 111111111111111111111111111111111111111111111111
--X-X 111111111111111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      111111111111111111111111111111111111111111111111
      000000000111111111122222222233333333334444444445
      123456789012345678901234567890123456789012345678
```

```
XXXXX 777777777777777777777777777777777799999999999999
XXXX- 555555555555555544333333222222222244444444443333
-XXXX 222222221111111111111111111111111111111111111111
XXX-X 666666666666666666666666666666668888888888888888
X-XXX 777777777777777777777777777777777799999999999999
XX-XX 222222222222222222222222222222224444444444444444
-XXX- 222222111111111111111111111111112222222222222222
XXX-- 555555555555555555555555555555557777777777777777
--XXX 333333222222222222222222222222224444444444444444
X-XX- 111111111111111111111111111111111333333333333333
-XX-X 111111111111111111111111111111111333333333333333
XX-X- 444444444444444444444444444444446666666666666666
-X-XX 222222222222222222222222222222224444444444444444
XX--X 666666666666666666666666666666668888888888888888
X--XX 777777777777777777777777777777777799999999999999
X-X-X 777777777777777777777777777777777799999999999999
-XX-- 333333333333333333333333333333335555555555555555
--XX- 333333333333333333333333333333335555555555555555
XX--- 444444444444444444444444444444446666666666666666
---XX 444444444444444444444444444444446666666666666666
-X-X- 555555555555555555555555555555557777777777777777
X-X-- 666666666666666666666666666666668888888888888888
--X-X 666666666666666666666666666666668888888888888888
X--X- 777777777777777777777777777777777799999999999999
-X--X 777777777777777777777777777777777799999999999999
X---X 777777777777777777777777777777777799999999999999
--X-- 777777777777777777777777777777777799999999999999
-X--- 777777777777777777777777777777777799999999999999
---X- 777777777777777777777777777777777799999999999999
X---- 777777777777777777777777777777777799999999999999
----X 777777777777777777777777777777777799999999999999
----- 777777777777777777777777777777777799999999999999

      111111111111111111111111111111111111111111111111
      000000000111111111122222222223333333333444444444 5
      123456789012345678901234567890123456789012345678
```

Figure 4.8(6)c continued

```
XXXXX 000000000000000000000000000000000000000000000000
XXXX- 666666666666666666666666677777777777777777788888
-XXXX 888888888888888888888888888888888888888888888888
XXX-X 111111111111111111111111111111111111111111111111
X-XXX 000000000000000000000000000000000000000000000000
XX-XX 555555555555555555555555555555555555555555555555
-XXX- 777777777777777777777777777777777777777777777777
XXX-- 222222222222222225555555566666666666666666666666
--XXX 555555555555555555566666666666666666666666666666
X-XX- 666666666666666666666666666666666666666666666666
-XX-X 666666666666666666666666666666666666666666666666
XX-X- 333333333333333333333333444444444444444444445555
-X-XX 555555555555555555555555555555555555555555555555
XX--X 111111111111111111111111111111111111111111111111
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 444444444444444444444444444444444444444444444444
--XX- 444444444444444444444444444444444444444444444444
XX--- 333333333333333333333333333333333333333333333333
---XX 333333333333333333333333333333333333333333333333
-X-X- 222222222222222222222222222222222222222222222222
X-X-- 111111111111111111111111111111111111111111111111
--X-X 111111111111111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      111111111111111111111111111111111111111111111112
      555555555566666666667777777777788888888889999999990
      123456789012345678901234567890123456789012345678901
```

```
XXXXX 999999999999999999999999999999999999999999999999
XXXX- 333333333333333333333333222222222222222221111111
-XXXX 111111111111111111111111111111111111111111111111
XXX-X 888888888888888888888888888888888888888888888888
X-XXX 999999999999999999999999999999999999999999999999
XX-XX 444444444444444444444444444444444444444444444444
-XXX- 222222222222222222222222222222222222222222222222
XXX-- 777777777777777774444444333333333333333333333333
--XXX 444444444444444444333333333333333333333333333333
X-XX- 333333333333333333333333333333333333333333333333
-XX-X 333333333333333333333333333333333333333333333333
XX-X- 666666666666666666666665555555555555555555554444
-X-XX 444444444444444444444444444444444444444444444444
XX--X 888888888888888888888888888888888888888888888888
X--XX 999999999999999999999999999999999999999999999999
X-X-X 999999999999999999999999999999999999999999999999
-XX-- 555555555555555555555555555555555555555555555555
--XX- 555555555555555555555555555555555555555555555555
XX--- 666666666666666666666666666666666666666666666666
---XX 666666666666666666666666666666666666666666666666
-X-X- 777777777777777777777777777777777777777777777777
X-X-- 888888888888888888888888888888888888888888888888
--X-X 888888888888888888888888888888888888888888888888
X--X- 999999999999999999999999999999999999999999999999
-X--X 999999999999999999999999999999999999999999999999
X---X 999999999999999999999999999999999999999999999999
--X-- 999999999999999999999999999999999999999999999999
-X--- 999999999999999999999999999999999999999999999999
---X- 999999999999999999999999999999999999999999999999
X---- 999999999999999999999999999999999999999999999999
----X 999999999999999999999999999999999999999999999999
----- 999999999999999999999999999999999999999999999999

      111111111111111111111111111111111111111111111112
      555555555566666666667777777777788888888889999999990
      123456789012345678901234567890123456789012345678901
```

Figure 4.8(6)d continued

```
XXXXX 000000000000000001116666677777777888888
XXXX- 000000000000000000000000011112222222222
-XXXX 000000000111111122222222222222224777777
XXX-X 000111111133333333333333333333333333333
X-XXX 000011111112444444457777777788888888888
XX-XX 000000000000000000000000000000000000000
-XXX- 003444445555555556666666667777777777777
XXX-- 000000000000011145555555555556666666666
--XXX 000000022444444444445555555556666666666666
X-XX- 000000001111111111111222222222222222333
-XX-X 000001111111111111111122222233566666666
XX-X- 000000001111111111111224555555555555555
-X-XX 000000000000000000000000000000000000000
XX--X 000000000000000000000000000000000000000
X--XX 000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000
-XX-- 133333333333334444444444444444444444455
--XX- 012333333333334444444444444444444444455
XX--- 022222222222223333333333333333333333344
---XX 000111122222222333333333333333333333334
-X-X- 011111111111112222222222222222222223333
X-X-- 000000000000011111111111111111111111111
--X-X 000000000000011111111111111111111112222
X--X- 000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000011111
X---X 000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000

      000000000000000000000000000000111111111
      000000111111111122333334666677880223479 99
      234589012345678180123751679044776790806 7
```

↑

WA.140

↓

```
XXXXX 244555556666666667667222221222222211111 1
XXXX- 244555556666666667778888887888777777777
-XXXX 244555556555555554555666666677777752222 2
XXX-X 244444445533333334445555555666666666666
X-XXX 244544445554222223321111111111111111111
XX-XX 244555556666666667778888888999999999999
-XXX- 241111111111111111112222222222222222222
XXX-- 244555556666655522233333334333333333333
--XXX 244555332222222223223333332333333333333
X-XX- 244555554555555555666676666667777777766 6
-XX-X 244554445555555555666776666776643333333
XX-X- 244555555555555555666776643344444444444
-X-XX 244555556666666667778888888999999999999
XX--X 244555556666666667778888888999999999999
X--XX 244555556666666667778888888999999999999
X-X-X 244555556666666667778888888999999999999
-XX-- 111222222333333322333444444455555555555 44
--XX- 232222222333333322333444444455555555555 44
XX--- 222333334444444334445555555666666666666 55
---XX 244444443444444433444555555666666666666 5
-X-X- 233444444555555444555666666667777777776 666
X-X-- 244555556666666555666777777778888888888 888
--X-X 244555556666666555666777777778888888887 777
X--X- 244555556666666667778888888999999999999
-X--X 244555556666666667778888888999999998888
X---X 244555556666666667778888888999999999999
--X-- 244555556666666667778888888999999999999
-X--- 244555556666666667778888888999999999999
---X- 244555556666666667778888888999999999999
X---- 244555556666666667778888888999999999999
----X 244555556666666667778888888999999999999
----- 244555556666666667778888888999999999999

      000000000000000000000000000000111111111
      000000111111111122333334666677880223479 99
      234589012345678180123751679044776790806 7
```

Figure 4.9(1) Condensed Line Values (WA)

```
XXXXX  000000000000000000000000000000000223
XXXX-  000000000000000001111111111111111111
-XXXX  000000000000000000000000000000000000
XXX-X  000000000002222222222222222222222222
X-XXX  000000000000001122222222455667777777
XX-XX  000000000000000000000000000000000000
-XXX-  001113444444444444444444445666666666
XXX--  000000001111111111111111222222222222
--XXX  001111111111111111111111111122455555
X-XX-  000000000111233444444445666666666666
-XX-X  000111122222222222222222222222222222
XX-X-  011111111111111111111111111111113355
-X-XX  000000000000111111122333555555555555
XX--X  000000000000000000000000000000000000
X--XX  000000000000000000000000000000000000
X-X-X  000000000000000000000000000000000000
-XX--  333333333333333333333444444444444444
--XX-  111111111233333333333444444444444444
XX---  000022222222222222222233333333333333
---XX  222222222222222222222333333333333333
-X-X-  111111111111111111112222222222222222
X-X--  000000000000000000000000001111111111
--X-X  000000000000000001111111111111111111
X--X-  000000000000000000000000000000000000
-X--X  000000000000000000000000000001111111
X---X  000000000000000000000000000000000000
--X--  000000000000000000000000000000000000
-X---  000000000000000000000000000000000000
---X-  000000000000000000000000000000000000
X----  000000000000000000000000000000000000
----X  000000000000000000000000000000000000
-----  000000000000000000000000000000000000

       000000000000000000000000001111111112
       000011111222333444777777889901246678 90
       258914691234583680145691589665757974 0
```

```
XXXXX  444444533333333333333333677777788886 65
XXXX-  444444555555555555544444566666677777 77
-XXXX  444444555555555555555555677777788888 88
XXX-X  444444555555333333333334555555566666 66
X-XXX  444444555555555443333333432221111111 11
XX-XX  444444555555555555555555677777788888 88
-XXX-  443331111111111111111111233211122222 22
XXX--  444444554444444444444444555555566666 66
--XXX  443333344444444444444444566666564333 33
X-XX-  444444555444322111111111111111122222 22
-XX-X  444333433333333333333334555555566666 66
XX-X-  433333444444444444444445666666677755 33
-X-XX  444444555555444444443322332222233333 33
XX--X  444444555555555555555555677777788888 88
X--XX  444444555555555555555555677777788888 88
X-X-X  444444555555555555555555677777788888 88
-XX--  111111222222222222222211233333334444 44
--XX-  333333444432222222222211233333334444 44
XX---  444422333333333333333332344444445555 55
---XX  222222333333333333333223444444555555 5
-X-X-  333333444444444444444333345555556666 66
X-X--  444444555555555555555555677776677777 77
--X-X  444444555555555555444444566666677777 77
X--X-  444444555555555555555555677777788888 88
-X--X  444444555555555555555555677777787777 77
X---X  444444555555555555555555677777788888 88
--X--  444444555555555555555555677777788888 88
-X---  444444555555555555555555677777788888 88
---X-  444444555555555555555555677777788888 88
X----  444444555555555555555555677777788888 88
----X  444444555555555555555555677777788888 88
-----  444444555555555555555555677777788888 88

       000000000000000000000000001111111112
       000011111222333444777777889901246678 90
       258914691234583680145691589665757974 0
```

Figure 4.9(2) Condensed Line Values (W6)

```
XXXXX 000000000000000000001111111111111111112233333333
XXXX- 000000000000000000002222222222222244777838888889
-XXXX 000012223355555666666666666677777777777788888889
XXX-X 000111111111111111111111111111111111333333333333
X-XXX 000011222222222222222222222222222222333333333333
XX-XX 000000000000000000001111111111111111111111111111
-XXX- 000000000001111112344444466677777777777777777788
XXX-- 000000000000025555555555555566666666666666666666
--XXX 033344444444445555555555555555555555555555567777
X-XX- 011111111112222222222222334455666666666666666666
-XX-X 000000000000000000000011222222222233333345555555
XX-X- 000000000000000000000000000000000122222222222233
-X-XX 000001111133333355555555555555555555555555555555
XX--X 000000022222222222222222222222222333333333333555
X--XX 000000000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000000000
-XX-- 223333333333444444444444444444444444444444444444
--XX- 223333333333444444444444444444444444444444444444
XX--- 112222222222222222222233333333333333333333333333
---XX 111111222222333333333333333333333333333333333333
-X-X- 001111111112222222222222222222222222222222222222
X-X-- 000000000000000000000000000000001111111111111111
--X-X 000000000111111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000000

      000000000000000000000000000000001111111111111111
      000001111222222233334445567888900011122225588999 9
      245671246234567801674596788014223678901287801234 5
```

```
XXXXX 344455555566666777777666666677777777777676666667
XXXX- 344455555556666677777555555556666666441111111111
-XXXX 344443332211111111111111111111111111111111111111
XXX-X 344344444455555666666666666677777777555556666667
X-XXX 344444333344444555555555555566666666666655666667
XX-XX 344455555556666677777777666667777777777788888889
-XXX- 344455555556555566543333331111111111111112222212
XXX-- 344455555556666412222222222221222222222223333334
--XXX 311111111122221222222222222233333333333334444322
X-XX- 333344444454444555555555554433332222222223333334
-XX-X 344455555566666777777777766556666666665555665444
XX-X- 344455555566666777777777777788888876666667777767
-X-XX 344454444433333422222222222233333333333344444445
XX--X 344455533344444555555555555666665555555566666445
X--XX 344455555566666777777777777788888888888899999998
X-X-X 344455555566666777777777777788888888888899999998
-XX-- 121122222233222333333333333344444444444455555556
--XX- 121122222233222333333333333344444444444455555556
XX--- 232233333344444555555544444455555555555666666667
---XX 233344333344433344444444444455555555555666666667
-X-X- 343344444454444555555555555566666666666677777778
X-X-- 344455555566666777777777777788777777777788888889
--X-X 344455555545555566666666666677777777777788888889
X--X- 344455555556666677777777777788888888888899999998
-X--X 344455555556666677777777777788888888888899999998
X---X 344455555556666677777777777788888888888899999998
--X-- 344455555556666677777777777788888888888899999998
-X--- 344455555556666677777777777788888888888899999998
---X- 344455555556666677777777777788888888888899999998
X---- 344455555556666677777777777788888888888899999998
----X 344455555556666677777777777788888888888899999998
----- 344455555556666677777777777788888888888899999998

      000000000000000000000000000000001111111111111111
      000001111222222233334445567888900011122225588999 9
      245671246234567801674596788014223678901287801234 5
```

Figure 4.9(3) Condensed Line Values (W8)

```
XXXXX 0000000000111111111122222222222
XXXX- 0000000011111111111444445777777
-XXXX 0000000000000000000001111111111
XXX-X 0000000011111111111111122222233
X-XXX 0001111111222222444555555555555
XX-XX 0000000000000000000000000000000
-XXX- 0011111111111111112222222222222
XXX-- 0000000000000011111111111111111
--XXX 0004444444555566666666666666666
X-XX- 0112222222225555555555555555555
-XX-X 0011111111111111111112222222355
XX-X- 0000000000000000000111111122222
-X-XX 0000000000000011222222222222234
XX--X 0000000000000000000000000000000
X--XX 0000000000000000000000000000000
X-X-X 0000000000000000000000000000000
-XX-- 1111334444444444444444444444444
--XX- 3333334444444444444444444444444
XX--- 2222223333333333333333333333333
---XX 0001111333333333333333333333333
-X-X- 1111112222222222222222222222222
X-X-- 0000000000000000000000000001111
--X-X 0000011111111111111111111111111
X--X- 0000000000000000000000000000000
-X--X 0000000000000000000000000000000
X---X 0000000000000000000000000000000
--X-- 0000000000000000000000000000000
-X--- 0000000000000000000000000000000
---X- 0000000000000000000000000000000
X---- 0000000000000000000000000000000
----X 0000000000000000000000000000000
----- 0000000000000000000000000000000

      0000000000000011111111111111111
      0000011123333400111334566777899 99
      2467934860149059234074404279412 34
```

```
XXXXX 4445555555665566666666655556666666
XXXX- 4445555554555566666663333321111111
-XXXX 4445555555666677777777776667777777
XXX-X 4445555554555566666666665566665555
X-XXX 4444444445444555333222222233333333
XX-XX 4445555555666677777777777778888888
-XXX- 4434444444555566666555555556666666
XXX-- 4445555555666676666666666667777777
--XXX 4441111111111111111111111112222222
X-XX- 4333333334441222222222222233333333
-XX-X 4434444444555566666666555555663333
XX-X- 4445555555666677777777666666766666
-X-XX 4445555553666677665555555556666654
XX--X 4445555555666677777777777778888888
X--XX 4445555555666677777777777778888888
X-X-X 4445555555666677777777777778888888
-XX-- 3334221112222333333333333344444444
--XX- 1112221112222333333333333344444444
XX--- 2223332223333444444444444455555555
---XX 4444444223333444444444444455555555
-X-X- 3334443334444555555555555666666666
X-X-- 4445555556666677777777777778777777
--X-X 4445544445555666666666666677777777
X--X- 4445555556666677777777777778888888
-X--X 4445555556666677777777777778888888
X---X 4445555556666677777777777778888888
--X-- 4445555556666677777777777778888888
-X--- 4445555556666677777777777778888888
---X- 4445555556666677777777777778888888
X---- 4445555556666677777777777778888888
----X 4445555556666677777777777778888888
----- 4445555556666677777777777778888888

      0000000000000011111111111111111111
      0000011123333400111334566777899 99
      2467934860149059234074404279412 34
```

Figure 4.9(4) Condensed Line Values (WC)

```
XXXXX 000000000000000000000000000000000000000000000
XXXX- 000000011133333333333333333344455555555556677777
-XXXX 000000011111111111111111111111111111111111112
XXX-X 000000000000000011122222222222222333333333344
X-XXX 000000023444444444555555555556666667777777777
XX-XX 000000000000000000000000000000000000000000000
-XXX- 000013444444444555555555555555555555555677777
XXX-- 000000111111234444456666666666666666666666666
--XXX 000000000001111111111111111222222222345555555
X-XX- 001122333334444455555555555556666666666666666
-XX-X 000000000001111111111122222222345555555555666
XX-X- 000000000011111111111222222222222234444445555
-X-XX 000000112223333344444444455555555555555555555
XX--X 000000000000000000000000000000000000000000000
X--XX 000111111111111111111111111111111111111111111
X-X-X 000000000000000000000000000000000000000000000
-XX-- 333333333333333333333333444444444444444444444
--XX- 333333333333333333333333444444444444444444444
XX--- 122222222222222222222222333333333333333333333
---XX 222222222222222222222222333333333333333333333
-X-X- 111111111111111111111122222222222222222222222
X-X-- 000000000000000000001111111111111111111111111
--X-X 000000000000000000001111111111111111111111111
X--X- 000000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000000

      000000000000000000000000000000001111111111111
      000001111112233334444555666677899001133444588 8
      235890134562724570134015125646578571201267013 6
```

```
XXXXX 444444555555555666667777777777777788888888888
XXXX- 444444544422222333334444444333322222333332211111
-XXXX 444444544444444555556666666666666667777777777776
XXX-X 444444555555555566555555555555555554555555555544
X-XXX 444444532111111222112222222221111111111111111111
XX-XX 444444555555555566666777777777777778888888888888
-XXX- 444431111111111111111122222222222222233333211111
XXX-- 444444444444321222211111111111111112222222222222
--XXX 444444333354444555556666666665555555566543333333
X-XX- 443322222221111211112222222222211111222222222222
-XX-X 444444555554444555555666555555555543223333333222
XX-X- 444444555544444555555665555555555555565444443333
-X-XX 444444443332222322222333333222222222333333333333
XX--X 444444555555555566666777777777777777788888888888
X--XX 444333444444444555556666666666666667777777777777
X-X-X 444444555555555566666777777777777777788888888888
-XX-- 111111222222222333334444433333333333444444444444
--XX- 111111222222222333334444433333333333444444444444
XX--- 322222333333333444445555544444444444555555555555
---XX 222222333333333444445555544444444444555555555555
-X-X- 333333344444444455555665555555555555666666666666
X-X-- 444444555555555566666766666666666666677777777777
--X-X 444444555555555566666766666666666666677777777777
X--X- 444444555555555566666777777777777777788888888888
-X--X 444444555555555566666777777777777777788888888888
X---X 444444555555555566666777777777777777788888888888
--X-- 444444555555555566666777777777777777788888888888
-X--- 444444555555555566666777777777777777788888888888
---X- 444444555555555566666777777777777777788888888888
X---- 444444555555555566666777777777777777788888888888
----X 444444555555555566666777777777777777788888888888
----- 444444555555555566666777777777777777788888888888

      000000000000000000000000000000001111111111111
      000001111112233334444555666677899001133444588 8
      235890134562724570134015125646578571201267013 6
```

Figure 4.9(5) Condensed Line Values (W4)

```
XXXXX 000000000000000000000000000000000000000000
XXXX- 000000001111111111111222222222223455666678 8
-XXXX 000000000000000111112225555555556666888888 8
XXX-X 000000000000000000000000000001111111111111
X-XXX 000000000000000000000000000000000000000000
XX-XX 003333355555555555555555555555555555555555
-XXX- 000111111111111111112222222255566666777777 7
XXX-- 000000000000000001111111111222222225566 6
--XXX 000000000000000000000000004444555555555666 6
X-XX- 002333344444555555555555666666666666666666
-XX-X 001111111133344555555555666666666666666666
XX-X- 000001111111111111111111111113333333333444 5
-X-XX 000011111111223333333344555555555555555555
XX--X 000000000000000011111111111111111111111111
X--XX 000000000000000000000000000000000000000000
X-X-X 000000000000000000000000000000000000000000
-XX-- 333333333334444444444444444444444444444444
--XX- 333333333334444444444444444444444444444444
XX--- 122222222223333333333333333333333333333333
---XX 222222222223333333333333333333333333333333
-X-X- 111111111222222222222222222222222222222222
X-X-- 000000000000000000000000000011111111111111
--X-X 000000111111111111111111111111111111111111
X--X- 000000000000000000000000000000000000000000
-X--X 000000000000000000000000000000000000000000
X---X 000000000000000000000000000000000000000000
--X-- 000000000000000000000000000000000000000000
-X--- 000000000000000000000000000000000000000000
---X- 000000000000000000000000000000000000000000
X---- 000000000000000000000000000000000000000000
----X 000000000000000000000000000000000000000000
----- 000000000000000000000000000000000000000000

      000000000000000000000000000000111111111111
      000001111122222223333445677899900112346677 9 9
      237892456012678923490492474059779683448957 45
```

```
XXXXX 444444466666666666666677777777777799999999
XXXX- 444444465555555555555444555555555432433332 11
-XXXX 444444466666666555554441222222211111111111
XXX-X 444444466666666666666677777666666688888888
X-XXX 444444466666666666666677777777777799999999
XX-XX 441111111111111111111122222222222444444444
-XXX- 444333355555555555554444455522211111222222 2
XXX-- 444444466666666666555555666665555555774433 3
--XXX 444444466666666666666677333322222444333333
X-XX- 442111122222111111111111111111111133333333
-XX-X 443333355533322211111111211111111111333333 33
XX-X- 444443355555555555555555556666444444446666 5554
-X-XX 444433355555544333333332222222222224444444 4
XX--X 444444466666666665555555666666666668888888 8
X--XX 444444466666666666666666677777777777799999999
X-X-X 444444466666666666666666677777777777799999999
-XX-- 111111133332222222222222333333333333555555 55
--XX- 111111133332222222222222333333333333555555 55
XX--- 322222244443333333333333344444444444466666666
---XX 222222244443333333333333344444444444466666666
-X-X- 333333335544444444444444555555555555777777 77
X-X-- 444444466666666666666667777666666668888888
--X-X 444444355555555555555555666666666668888888
X--X- 444444466666666666666667777777777799999999
-X--X 444444466666666666666667777777777799999999
X---X 444444466666666666666667777777777799999999
--X-- 444444466666666666666667777777777799999999
-X--- 444444466666666666666667777777777799999999
---X- 444444466666666666666667777777777799999999
X---- 444444466666666666666667777777777799999999
----X 444444466666666666666667777777777799999999
----- 444444466666666666666667777777777799999999

      000000000000000000000000000000111111111111
      000001111122222223333445677899900112346677 9 9
      237892456012678923490492474059779683448957 45
```

Figure 4.9(6) Condensed Line Values (WE)

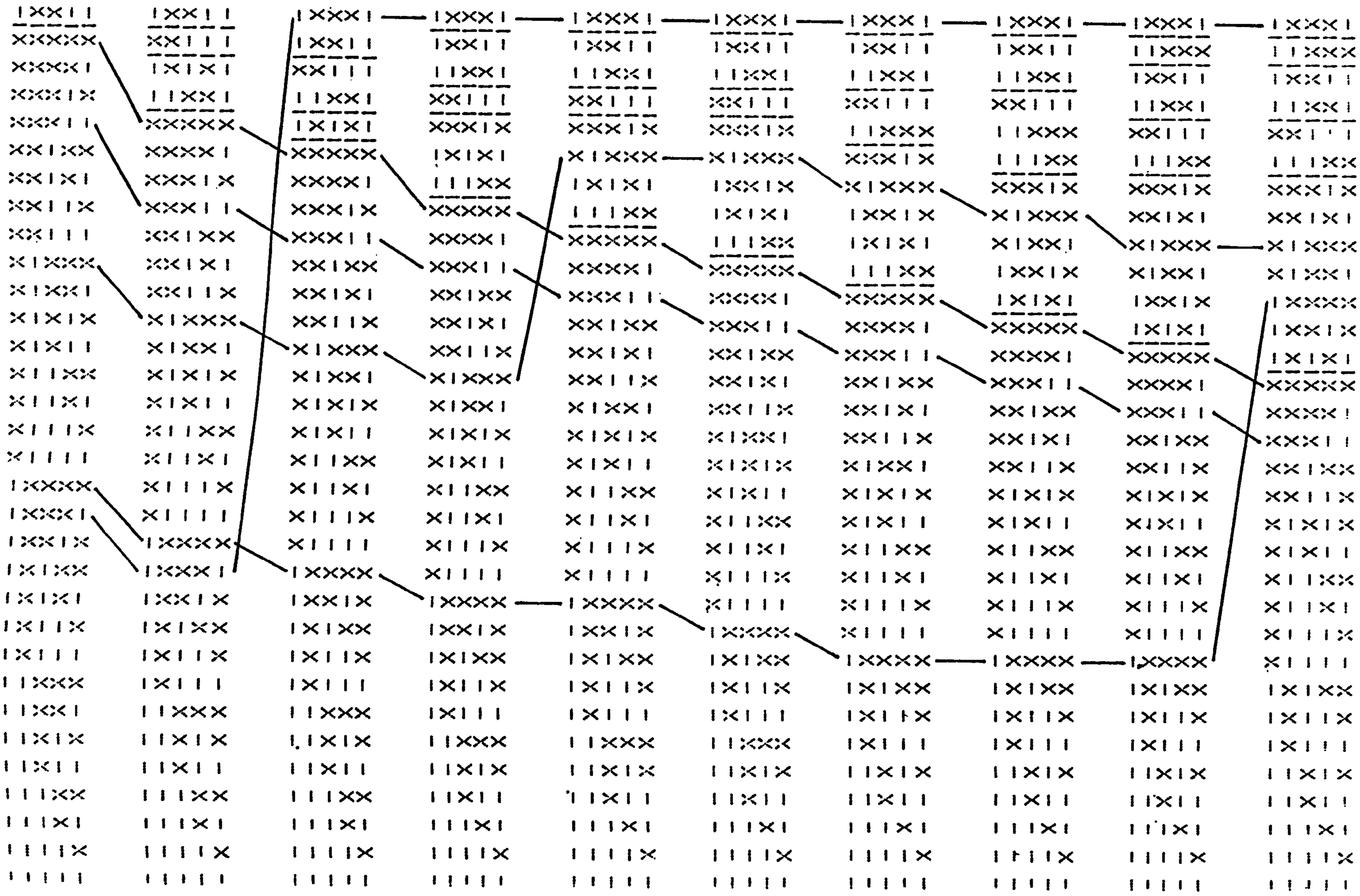


Figure 4.10(1) Line Rank Tables (WA)

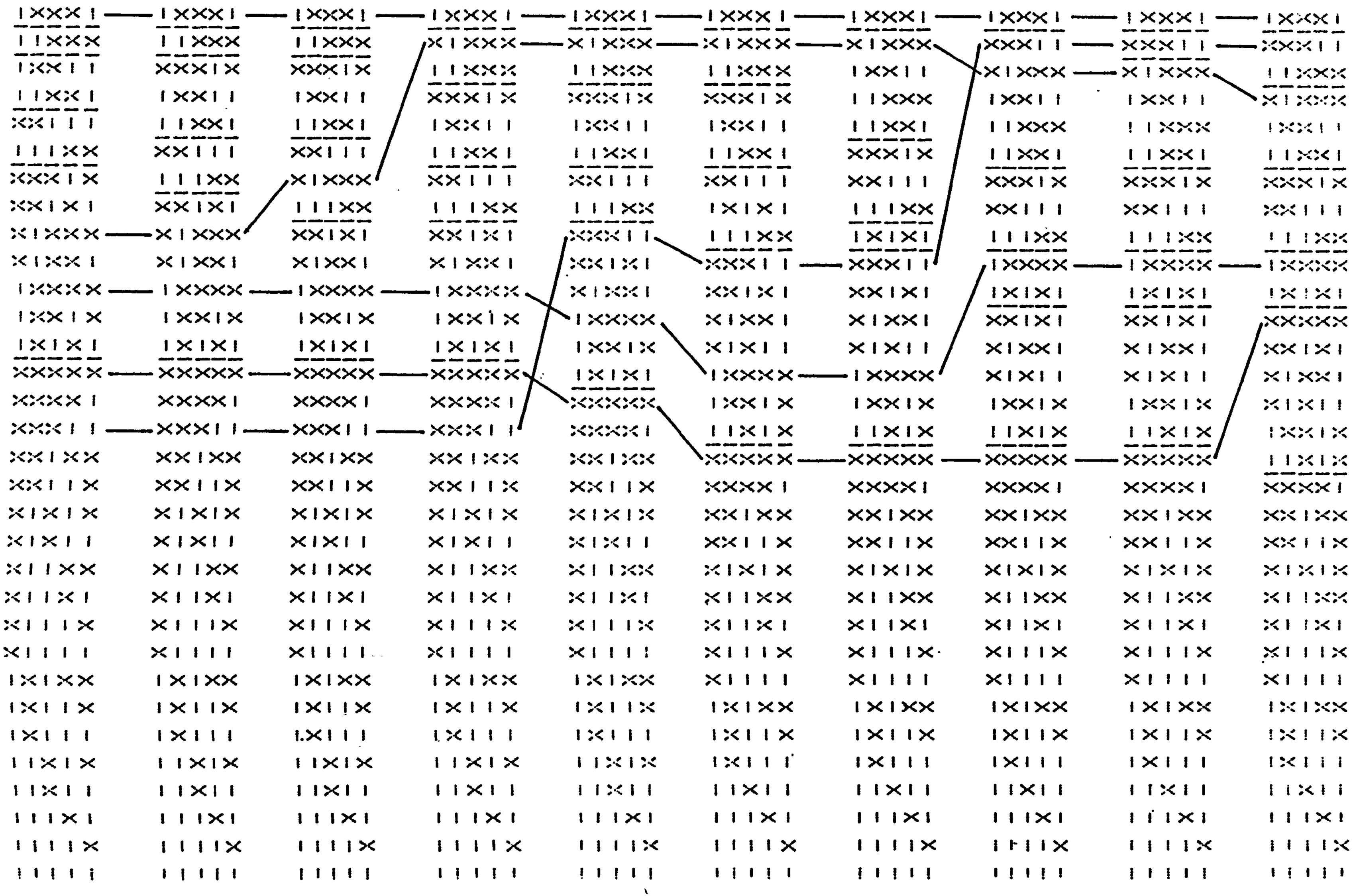


Figure 4.10(1) continued

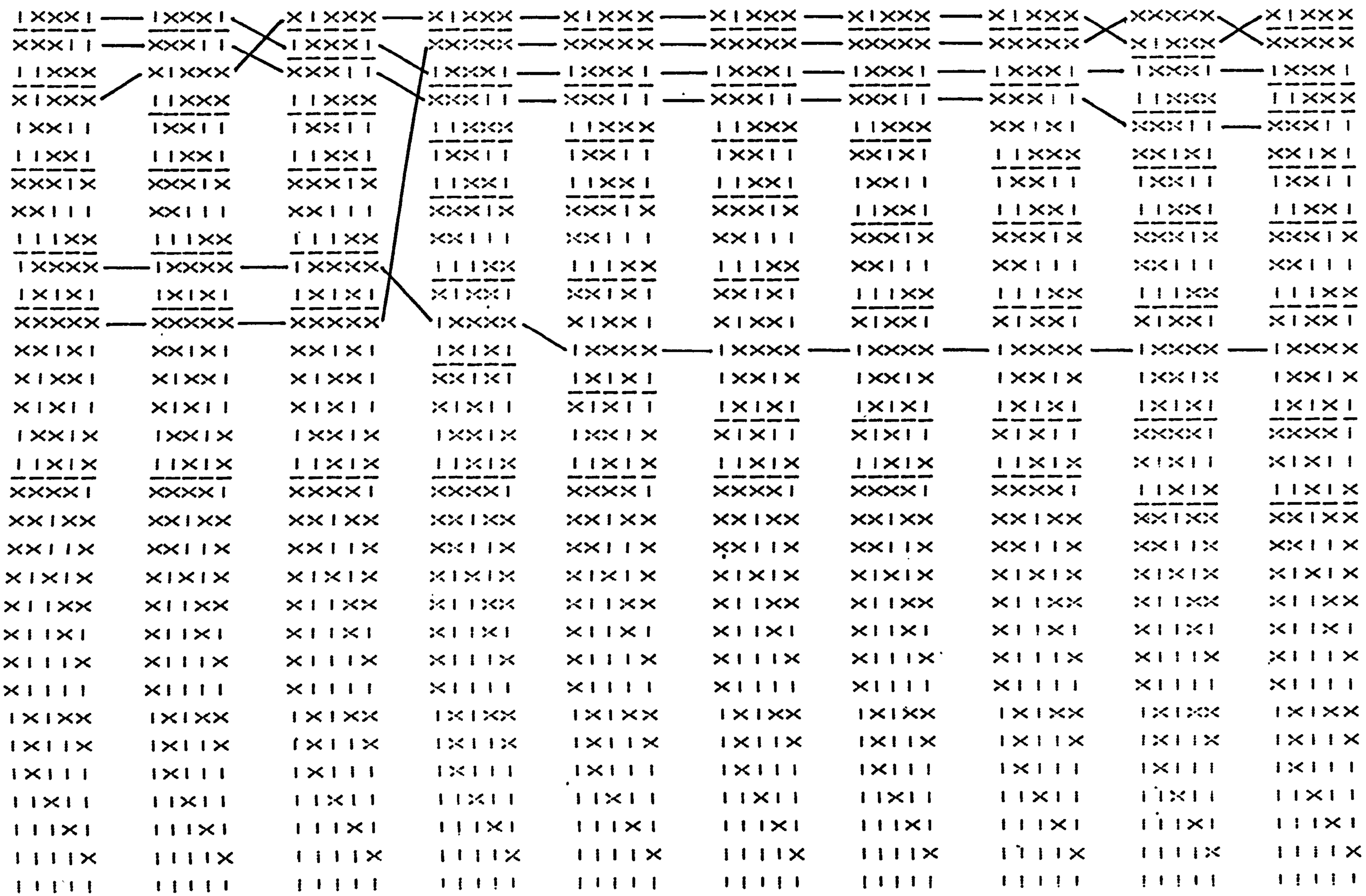


Figure 4.10(1) continued

WA.140

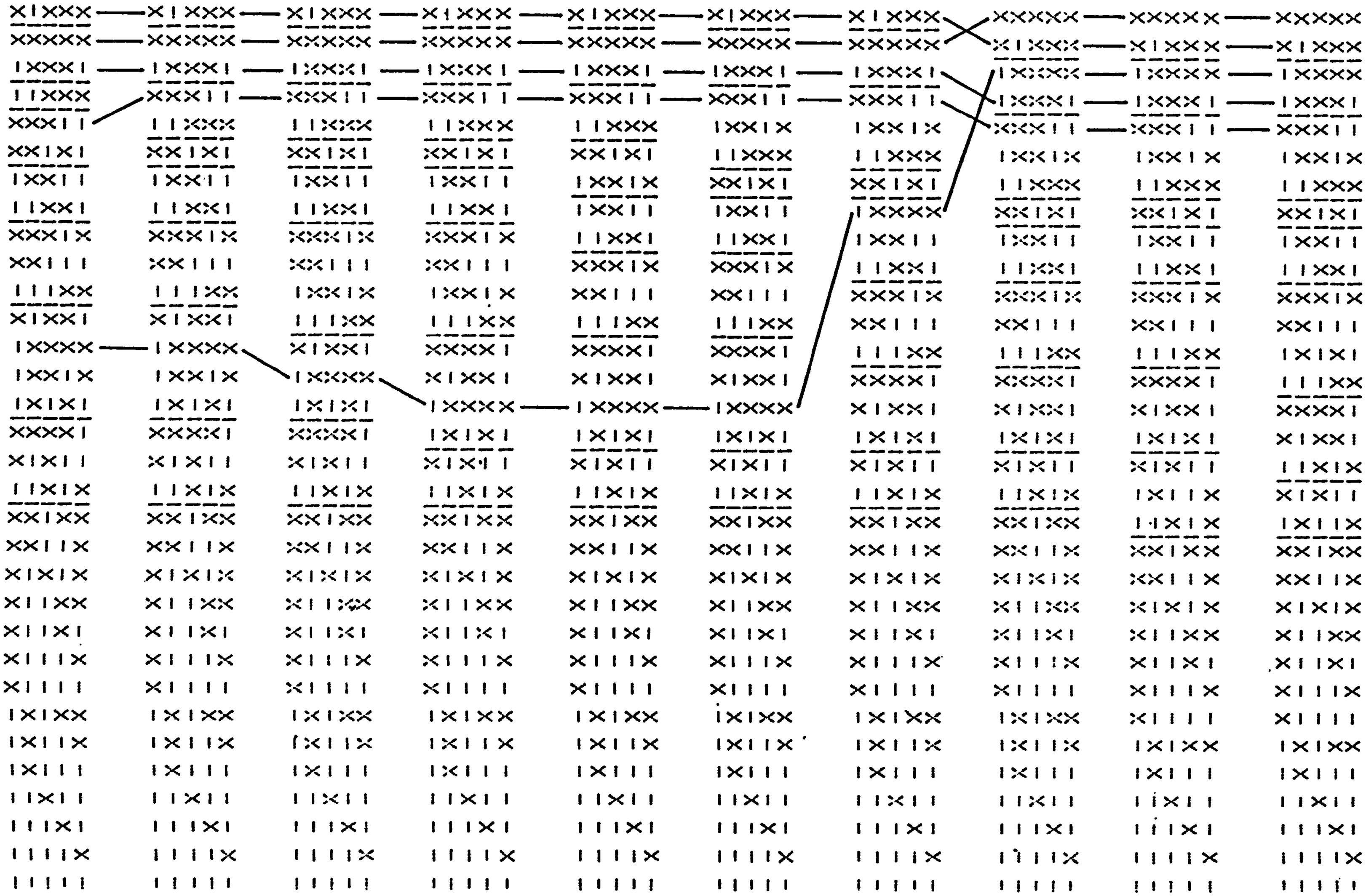


Figure 4.10(1) continued

```
XXXXX——XXXXX——XXXXX——XXXXX
X1XXX——X1XXX——X1XXX——X1XXX
1XXXX——1XXXX——1XXXX——1XXXX
1XXX1——1XXX1——1XXX1——1XXX1
XXX11——XXX11——XXX11——XXX11
1XX1X    1XX1X    1XX1X    1XX1X
11XXX    11XXX    11XXX    11XXX
XX1X1    XX1X1    XX1X1    XX1X1
1XX11    1XX11    1XX11    1XX11
11XX1    11XX1    11XX1    11XX1
XXX1X    XXX1X    XX111    XX111
XX111    XX111    XXX1X    111XX
1X1X1    X1XX1    X1XX1    XXX1X
111XX    1X1X1    1X1X1    X1XX1
XXXX1    111XX    111XX    1X1X1
X1XX1    XXXX1    XXXX1    XXXX1
11X1X    11X1X    11X1X    11X1X
X1X11    X1X11    X1X11    X1X11
1X11X    1X11X    1X11X    1X11X
XX1XX    XX1XX    XX1XX    XX1XX
XX11X    XX11X    XX11X    XX11X
X1X1X    X1X1X    X1X1X    X1X1X
X11XX    X11XX    X11XX    X11XX
X11X1    X11X1    X11X1    X11X1
X111X    X111X    X111X    X111X
X1111    X1111    X1111    X1111
1X1XX    1X1XX    1X1XX    1X1XX
1X111    1X111    1X111    1X111
11X11    11X11    11X11    11X11
111X1    111X1    111X1    111X1
1111X    1111X    1111X    1111X
11111    11111    11111    11111
```

Figure 4.10(1) continued

1XX11	1XX11	1XX11	1XX11	1XX11	1XXX1	1XXX1	1XXX1	1XXX1	1XXX1
111XX	111XX	111XX	111XX	XX111	1XX11	1XX11	1XX11	1XX11	1XX11
1X1X1	XX1X1	XX1X1	XX1X1	111XX	XX111	XX111	XX111	XX111	XX111
11XX1	1X1X1	1XXX1	1XXX1	XX1X1	111XX	111XX	1XX1X	1XX1X	1XX1X
XXXXX	11XX1	1X1X1	1XX1X	1XXX1	XX1X1	XX1X1	111XX	111XX	11XX1
XXXX1	XXXXX	11XXX	1X1X1	1XX1X	1XX1X	1XX1X	XX1X1	XXX11	111XX
XXX1X	XXXX1	11XX1	11XXX	1X1X1	1X1X1	1X1X1	1X1X1	XX1X1	XXX11
XXX11	XXX1X	XXXXX	11XX1	11XXX	11XXX	11XXX	11XXX	1X1X1	XX1X1
XX1XX	XXX11	XXXX1	XXXXX	11XX1	11XX1	11XX1	11XX1	11XXX	X1XX1
XX1X1	XX1XX	XXX1X	XXXX1	XXXXX	XXXXX	XXXXX	XXXXX	11XX1	1X1X1
XX111	XX111	XXX11	XXX1X	XXXX1	XXXX1	XXXX1	XXXX1	XXXXX	11XXX
XX111	XX111	XX1XX	XXX11	XXX1X	XXX1X	XXX1X	XXX1X	XXXX1	XXXXX
X1XXX	X1XXX	XX11X	XX1XX	XXX11	XXX11	XXX11	XXX11	XXX1X	XXXX1
X1XX1	X1XX1	XX111	XX11X	XX1XX	XX1XX	XX1XX	XX1XX	XX1XX	XXX1X
X1X1X	X1X1X	X1XXX	XX111	XX11X	XX11X	XX11X	XX11X	XX11X	XX1XX
X1X11	X1X11	X1XX1	X1XXX	X1XXX	X1XXX	X1XXX	X1XXX	X1XXX	XX11X
X11XX	X11XX	X1X1X	X1XX1	X1XX1	X1XX1	X1XX1	X1XX1	X1XX1	X1XXX
X111X	X111X	X11XX	X1X11	X1X11	X1X1X	X1X1X	X1X1X	X1X1X	X1X1X
X1111	X1111	X11X1	X11XX	X11XX	X1X11	X1X11	X1X11	X1X11	X1X11
1XXXX	1XXXX	X111X	X11X1	X11X1	X11XX	X11XX	X11XX	X11XX	X11XX
1XXX1	1XXX1	X1111	X111X	X111X	X11X1	X11X1	X11X1	X11X1	X11X1
1XX1X	1XX1X	1XXXX	X1111	X1111	X111X	X111X	X111X	X111X	X111X
1X1XX	1X1XX	1XX1X	1XXXX	1XXXX	X1111	X1111	X1111	X1111	X1111
1X111	1X111	1X11X	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX
11XXX	11XXX	1X111	1X1XX	1X1XX	1X1XX	1X1XX	1X1XX	1X1XX	1X1XX
11X1X	11X1X	11X1X	1X11X	1X11X	1X11X	1X11X	1X11X	1X11X	1X11X
11X11	11X11	11X11	1X111	1X111	1X111	1X111	1X111	1X111	1X111
111X1	111X1	111X1	111X1	111X1	111X1	111X1	111X1	111X1	111X1
1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X
11111	11111	11111	11111	11111	11111	11111	11111	11111	11111

Figure 4.10(2) Line Rank Tables (W6)

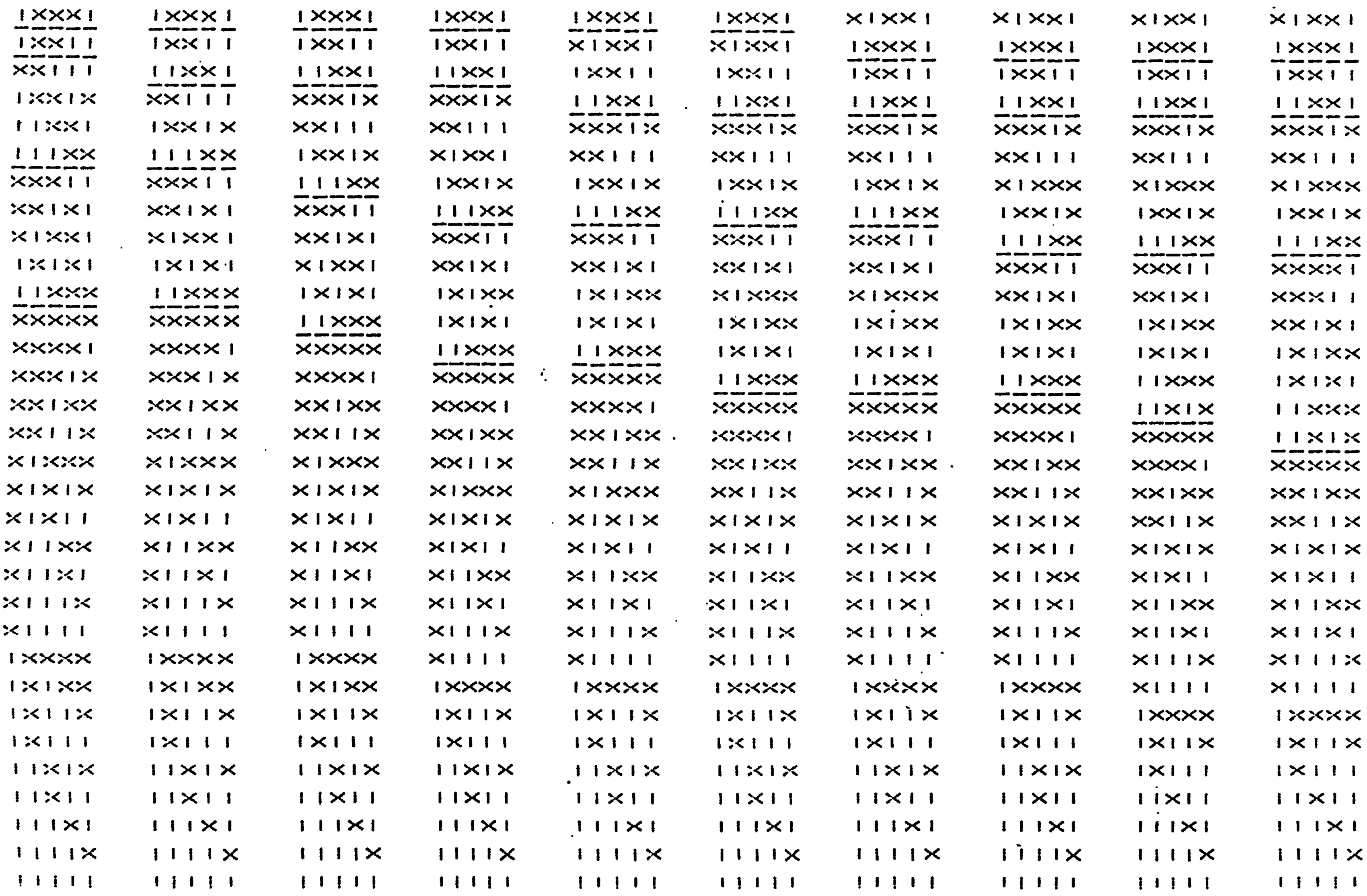


Figure 4.10(2) continued

```
XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI
IXXXI  IXXXI  IXXXI  IXXXI  IXXXI  IXXXI  IXIXX  XIXXX  XIXXX  IXXXI
IXXII  IXXII  IXXII  IXXII  IXXII  IXXII  XIXXX  IXIXX  IXXXI  XIXXX
IIXXI  IIXXI  IIXXI  IIXXI  IIXXI  IIXXI  IXXXI  IXXXI  IXIXX  IXIXX
XXXIX  XXXIX  XXXIX  IXIXX  XXIII  XXIII  IXXII  IXXII  IXXII  IXXII
XXIII  XXIII  XXIII  IIIXX  IXIXX  IXIXX  IIXXI  IIXXI  IIXXI  IIXXI
XIXXX  XIXXX  XIXXX  XXXIX  IIIXX  IIIXX  XXIII  XXIII  XXIII  XXIII
IXXIX  IXXIX  IXXIX  XXIII  XXXIX  XXXIX  IIIXX  IIIXX  IIIXX  IIIXX
IIIXX  IXIXX  IXIXX  XIXXX  XIXXX  XIXXX  XXXIX  XXXIX  XXXIX  XXXIX
XXXXI  IIIXX  IXIXI  IXXIX  IXXIX  IXXIX  XXXII  XXXII  XXXII  XXXII
XXXII  XXXXI  IIIXX  IXIXI  IXIXI  IXIXI  IXXIX  IXXIX  IXXIX  IXXIX
XXIXI  XXXII  XXXXI  XXXXI  XXXXI  XXXXI  IXIXI  IXIXI  IXIXI  IXIXI
IXIXX  XXIXI  XXXII  XXXII  XXXII  XXXII  XXXXI  XXXXI  XXXXI  XXXXI
IXIXI  IXIXI  XXIXI  XXIXI  XXIXI  XXIXI  XXIXI  XXIXI  XXIXI  XXIXI
IIXXX  IIXXX  IIXXX  IIXXX  IIXXX  IIXXX  IIXXX  IIXXX  IIXXX  IIXXX
IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX
XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX
XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX
XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX
XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX
XIXII  XIXII  XIXII  XIXII  XIXII  XIXII  XIXII  XIXII  XIXII  XIXII
XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX
XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI
XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX
XIIII  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII
IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX
IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX
IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII
IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII
IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI
IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX
IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII
```

Figure 4.10(2) continued

```
XIXXI  XIXXX  XIXXX  XIXXX  XIXXX  XIXXX  XIXXX  XIXXX  XIXXX  XIXXX
IXXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI
XIXXX  IXXXI  IXXXI  IXXXI  IXXXI  IXXXI  IXXXI  IXXXI  IXXXI  IXXXI
IXIXX  IXIXX  IXIXX  IXIXX  IXIXX  IXIXX  IXIXX  IXIXX  XXIXI  XXIXI
IXXII  IXXII  IXXII  IXXII  IXXII  IIXXX  IIXXX  IIXXX  IXIXX  IXIXX
IIXXI  IIXXI  IIXXI  IIXXI  IIXXX  IXXII  IXXII  IXXII  IIXXX  IIXXX
XXIII  XXIII  XXIII  XXIII  IIXXI  IIXXI  IIXXI  IIXXI  IXXII  IXXII
IIIXX  IIIXX  IIIXX  IIIXX  XXIII  XXIII  XXIXI  XXIXI  IIXXI  IIXXI
XXXIX  XXXIX  XXXIX  XXXIX  IIIXX  IIIXX  XXIII  XXIII  XXIII  XXXXX
XXXII  XXXII  XXXII  XXXII  XXXIX  XXXIX  IIIXX  IIIXX  IIIXX  XXIII
IXXIX  IXXIX  IXXIX  IXXIX  XXXII  XXXII  XXXIX  XXXXX  XXXXX  IIIXX
IXIXI  IXIXI  IXIXI  IXIXI  IXXIX  IXXIX  XXXII  XXXIX  XXXIX  XXXIX
XXXXI  XXXXI  IIXXX  IIXXX  IXIXI  IXIXI  IXXIX  XXXII  XXXII  XXXII
XXIXI  XXIXI  XXXXI  XXXXI  XXXXI  XXXXI  IXIXI  IXXIX  IXXIX  IXXIX
IIXXX  XIXII  XXIXI  XXIXI  XXIXI  XXIXI  XXXXI  IXIXI  IXIXI  IXIXI
IIXIX  IIXXX  XIXII  XIXII  XIXII  XIXII  XIXII  XXXXI  XXXXI  XXXXI
XXXXX  IIXIX  IIXIX  IIXIX  IXIIX  IXIIX  IXIIX  XIXII  XIXII  XIXII
XXIXX  XXXXX  XXXXX  XXXXX  IIXIX  IIXIX  IIXIX  IXIIX  IXIIX  IXIIX
XXIIX  XXIXX  XXIXX  XXIXX  XXXXX  XXXXX  XXXXX  IIXIX  IIXIX  IIXIX
XIXIX  XXIIX  XXIIX  XXIIX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX
XIXII  XIXIX  XIXIX  XIXIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX
XIIXX  XIIXX  XIIXX  XIIXX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX
XIIXI  XIIXI  XIIXI  XIIXI  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX
XIIIX  XIIIX  XIIIX  XIIIX  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI
XIIII  XIIII  XIIII  XIIII  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX
IXXXX  IXXXX  IXXXX  IXXXX  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII
IXIIX  IXIIX  IXIIX  IXIIX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX
IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII
IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII
IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI
IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX
IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII
```

Figure 4.10(2) continued

```
XIXXX
XIXXI

IXXXI
XXIXI

IXIXX

IIXXX
IXXII

IIXXI
XXXXX

XXIII

IIIXX
XXXIX

XXXII

IXXIX

IXIXI
XXXXI

XIXII

IXIIX

IIXIX
XXIXX

XXIIX

XIXIX

XIIXX

XIIXI

XIIIX

XIIII

IXXXX

IXIII

IIXII

IIIXI

IIIIX

IIIII
```

Figure 4.10(2) continued

```
IXXII   IIXXX   IXXII   IXXII   IIXXX   IIXXX   IIXXX   IIXXX   IIXXX   IIXXX
IIXXI   IXXII   IIXXX   IIXXX   IXXII   IXXII   IXXII   IXXII   IXXXX   IXXXX
XXIII   IIXXI   IIXXI   IIXXI   IIXXI   IIXXI   IIXXI   IIXXI   IXXII   IXXII
IIIXX   XXIII   XXIII   XXIII   XXIII   XXIII   XXIII   XXIIX   IIXXI   IIXXI
XXXXX   XIXXI   XIXXI   XXXIX   XXXIX   IXXXX   XIXXX   XXIII   XXIIX   XXIIX
XXXXI   IIIXX   IXIXI   XIXXI   XIXXX   XXXIX   IXXXX   XIXXX   XXIII   XXIII
XXXIX   XXXXX   IIIXX   IXIXI   XIXXI   XIXXX   IIIXX   IXXXX   XIXXX   XIXXX
XXXII   XXXXI   XXXXX   IIIXX   IXXXX   XIXXI   XXXIX   IIIXX   IIIXX   IIIXX
XXIXX   XXXIX   XXXXI   XXXXX   IXIXI   IXIXX   XIXXI   XXXIX   XXXIX   XXXIX
XXIXI   XXXII   XXXIX   XXXXI   IIIXX   IXIXI   IXIXX   XIXXI   XIXXI   XIXXI
XXIIX   XXIXX   XXXII   XXXII   XXXXX   IIIXX   IXIXI   IXIXX   IXIXX   IXIXX
XIXXX   XXIXI   XXIXX   XXIXX   XXXXI   XXXXX   XXXXX   IXIXI   IXIXI   IXIXI
XIXXI   XXIIX   XXIXI   XXIXI   XXXII   XXXXI   XXXXI   XXXXX   XXXXX   IIXIX
XIXIX   XIXXX   XXIIX   XXIIX   XXIXX   XXXII   XXXII   XXXXI   XXXXI   XXXXX
XIXII   XIXIX   XIXXX   XIXXX   XXIXI   XXIXX   XXIXX   XXXII   XXXII   XXXXI
XIIXX   XIXII   XIXIX   XIXIX   XXIIX   XXIXI   XXIXI   XXIXX   XXIXX   XXXII
XIIXI   XIIXX   XIXII   XIXII   XIXIX   XXIIX   XXIIX   XXIXI   XXIXI   XXIXX
XIIIX   XIIXI   XIIXX   XIIXX   XIXII   XIXIX   XIXIX   XIXIX   XIXIX   XXIXI
XIIII   XIIIX   XIIXI   XIIXI   XIIXX   XIXII   XIXII   XIXII   XIXII   XIXIX
IXXXX   XIIII   XIIIX   XIIIX   XIIXI   XIIXX   XIIXX   XIIXX   XIIXX   XIXII
IXXXI   IXXXX   XIIII   XIIII   XIIIX   XIIXI   XIIXI   XIIXI   XIIXI   XIIXX
IXXIX   IXXXI   IXXXX   IXXXX   XIIII   XIIIX   XIIIX   XIIIX   XIIIX   XIIXI
IXIXX   IXXIX   IXXXI   IXXXI   IXXXI   XIIII   XIIII   XIIII   XIIII   XIIIX
IXIXI   IXIXX   IXXIX   IXXIX   IXXIX   IXXXI   IXXXI   IXXXI   IXXXI   XIIII
IXIIX   IXIXI   IXIXX   IXIXX   IXIXX   IXXIX   IXXIX   IXXIX   IXXIX   IXXXI
IXIII   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXXIX
IIXXX   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII   IXIIX
IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IXIII
IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII
IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI
IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX
IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII
```

Figure 4.10(3) Line Rank Tables (W8)

```
11XXX  1XXXX  1XXXX  1XXXX  1XXXX  XXX11  1XXXX  1XXXX  1XXXX  1XXXX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1XXXX  11XXX  11XXX  1XX11  1XX11  1XXXX  XXX11  XXX11  XXX11  XXX11
1XX11  1XX11  1XX11  11XXX  11XXX  11XXX  11XXX  1X1XX  1X1XX  1X1XX
11XX1  1X1XX  1X1XX  11XX1  11XX1  -----  -----  -----  -----  -----
-----  -----  -----  -----  -----  1XX11  1XX11  11XXX  11XXX  11XXX
XX11X  11XX1  11XX1  1X1XX  1X1XX  11XX1  11XX1  -----  -----  -----
XX111  XX11X  XX11X  -----  -----  -----  -----  1XX11  1XX11  1XX11
X1XXX  XX111  XX111  111XX  111XX  1X1XX  1X1XX  11XX1  11XX1  11XX1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
111XX  XX111  XX111  XX11X  XXX11  111XX  111XX  111XX  111XX  1XXX1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XXX1X  X1XXX  X1XXX  XX111  XX11X  XX11X  XX11X  XX11X  XX11X  111XX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X1XX1  111XX  X1XX1  X1XXX  XX111  XX111  XX111  XX111  XX111  XX11X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1X1XX  XXX1X  1X1X1  X1XX1  X1XXX  X1XXX  X1XXX  X1XXX  X1XXX  XX111
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1X1X1  X1XX1  111XX  1X1X1  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1  X1XXX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1X1X1  1X1X1  XXX1X  XXX1X  1X1X1  1X1X1  1X1X1  1X1X1  1XXX1  X1XX1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
11X1X  11X1X  1XXX1  1XXX1  XXX1X  XXX1X  XXX1X  XXX1X  1X1X1  1X1X1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XXXXX  XXXXX  11X1X  11X1X  1XXXX  1XXXX  1XXXX  1XXXX  XXX1X  XXX1X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XXXXX  XXXXX  XXXXX  XXXXX  11X1X  11X1X  11X1X  11X1X  11X1X  11X1X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XXX11  XXX11  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX
XX1XX  XX1XX  XXX11  XXX11  XXXX1  XXXX1  XXXX1  XXXX1  XXXX1  XXXX1
XX1X1  XX1X1  XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  XX1XX
X1X1X  X1X1X  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1
X1X11  X1X11  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X
X11XX  X11XX  X1X11  X1X11  X1X11  X1X11  X1X11  X1X11  X1X11  X1X11
X11X1  X11X1  X11XX  X11XX  X11XX  X11XX  X11XX  X11XX  X11XX  X11XX
X111X  X111X  X11X1  X11X1  X11X1  X11X1  X11X1  X11X1  X11X1  X11X1
X1111  X1111  X1111  X1111  X1111  X1111  X1111  X1111  X1111  X1111
1XXX1  1XXX1  X1111  X1111  X1111  X1111  X1111  X1111  X1111  X1111
1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  1XX1X
1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X
1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111
11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11
111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1
1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X
11111  11111  11111  11111  11111  11111  11111  11111  11111  11111
```

Figure 4.10(3) continued

```
1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  XXX11
XXX11  XXX11  XXX11  XXX11  XXX11  XXX11  XXX11  1XXX1  1XXX1  1XXXX
1X1XX  1X1XX  1X1XX  1X1XX  1X1XX  1X1XX  1X1XX  XXX11  XXX11  1XXX1
11XXX  11XXX  11XXX  11XXX  11XXX  11XXX  11XXX  1X1XX  1X1XX  1X1XX
1XX11  1XXX1  1XXX1  1XXX1  1XXX1  1XXX1  1XXX1  11XXX  11XXX  11XXX
1XXX1  1XX11  1XX11  1XX11  1XX11  1XX11  1XX11  1XX11  1XX11  X1XX1
11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  1XX11
111XX  111XX  111XX  111XX  XXX11  XXX11  XXX11  XXX11  XXX11  11XX1
XX11X  XX11X  XXXX1  XXXX1  111XX  111XX  111XX  X1XX1  X1XX1  XX111
XX111  XX111  XX11X  XX11X  XXXX1  XXXX1  XXXX1  111XX  111XX  111XX
X1XXX  X1XXX  XX111  XX111  XX11X  XX11X  XX11X  XXXX1  XXXX1  XXXX1
X1XX1  X1XX1  X1XXX  X1XXX  X1XXX  X1XXX  X1XXX  XX11X  XX11X  XX11X
1X1X1  1X1X1  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1  X1XXX  X1XXX  X1XXX
XXX1X  XXX1X  1X1X1  1X1X1  1X1X1  1X1X1  1X1X1  1X1X1  1XX1X  1XX1X
11X1X  11X1X  XXX1X  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  1X1X1  1X1X1
XXXXX  XXXXX  11X1X  XXX1X  XXX1X  XXX1X  XXX1X  XXX1X  XXXXX  XXXXX
XXXX1  XXXX1  XXXXX  11X1X  11X1X  XX1XX  XX1XX  XX1XX  XXX1X  XXX1X
XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  11X1X  1XX1X  1XX1X  XX1XX  XX1XX
XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  11X1X  11X1X  11X1X  11X1X
X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  XX1X1  XX1X1  XX1X1  XX1X1
X1X11  X1X11  X1X11  X1X11  X1X11  X1X11  X1X1X  X1X1X  X1X1X  X1X1X
X11XX  X11XX  X11XX  X11XX  X11XX  X11XX  X1X11  X1X11  X1X11  X1X11
X11X1  X11X1  X11X1  X11X1  X11X1  X11X1  X11XX  X11XX  X11XX  X11XX
X111X  X111X  X111X  X111X  X111X  X111X  X11X1  X11X1  X11X1  X11X1
X1111  X1111  X1111  X1111  X1111  X1111  X111X  X111X  X111X  X111X
1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  X1111  X1111  X1111  X1111
1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X
1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111
11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11
111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1
1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X
11111  11111  11111  11111  11111  11111  11111  11111  11111  11111
```

Figure 4.10(3) continued

```
XXX11  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX
1XXXX  1XXX1  1XXX1  1XXX1  1XXX1  1XXX1  1XXX1  1XXX1  1XXX1  1XXX1
1XXX1  XXX11  XXX11  XXX11  XXX11  XXX11  XXX11  XXX11  XXX11  XXX11
1X1XX  1X1XX  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1
11XXX  11XXX  1X1XX  1X1XX  1X1XX  1X1XX  1X1XX  1X1XX  1X1XX  1X1XX
X1XX1  X1XX1  11XXX  11XXX  11XXX  11XXX  11XXX  11XXX  11XXX  11XXX
1XX11  1XX11  1XX11  1XX11  1XX11  1XX11  1XX11  1XX11  XXXX1  XXXX1
11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  1XX11  1XX11
XX111  XX111  XX111  XX111  XX111  XX11X  XX11X  XX11X  11XX1  11XX1
111XX  111XX  111XX  111XX  111XX  XX111  XX111  XX111  XXX1X  XXX1X
XXXX1  XXXX1  XXXX1  XXXX1  XXXX1  111XX  111XX  111XX  XX11X  XX11X
XX11X  XX11X  XX11X  XX11X  XX11X  XXXX1  XXXX1  XXXX1  XX111  XX111
X1XXX  X1XXX  X1XXX  X1XXX  X1XXX  X1XXX  X1XXX  XX1X1  111XX  1XX1X
1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  1XX1X  X1XXX  XX1X1  111XX
1X1X1  1X1X1  1X1X1  1X1X1  1X1X1  1X1X1  1X1X1  1XX1X  X1XXX  XX1X1
XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  1X1X1  1XX1X  X1XXX
XXX1X  XXX1X  XXX1X  XXX1X  XXX1X  XXX1X  XXX1X  XXXXX  1X1X1  1X1X1
XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  XXXX1  XXXXX  XXXXX
11X1X  11X1X  11X1X  X1X11  X1X11  X1X11  XX1X1  XX1XX  XX1XX  XX1XX
XX1X1  XX1X1  XX1X1  X1X11  X1X11  X1X11  X1X11  X1X11  X1X11  X1X11
X1X1X  X1X1X  X1X1X  11X1X  11X1X  11X1X  11X1X  11X1X  11X1X  11X1X
XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  X1X1X  X1X1X  X1X1X  X1X1X
X1X11  X1X11  X1X11  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X
X11XX  X11XX  X11XX  X11XX  X11XX  X11XX  X11XX  X11XX  X11XX  X11XX
X11X1  X11X1  X11X1  X11X1  X11X1  X11X1  X11X1  X11X1  X11X1  X11X1
X111X  X111X  X111X  X111X  X111X  X111X  X111X  X111X  X111X  X111X
X1111  X1111  X1111  X1111  X1111  X1111  X1111  X1111  X1111  X1111
1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X
1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111
11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11
111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1
1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X
11111  11111  11111  11111  11111  11111  11111  11111  11111  11111
```

Figure 4.10(3) continued

```
I XXXX   XXXXI   XXXXI   XXXXI   XXXXI   XXXXI   XXXXI   XXXXI   XXXXI   XXXXI
IXXXI    IXXXX   IXXXX   IXXXX   IXXXX   IXXXX   IXXXX   IXXXX   IXXXX   IXXXX
XXXII    IXXXI   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI
XIXXI    XXXII   XXXII   XXXII   XXXII   XXXII   XXXII   XXXII   XXXII   IIXXX
IXIXX    XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XXXII
IIXXX    IXIXX   IXIXX   IXIXX   IXIXX   IXIXX   IXIXX   IXXIX   IIXXX   XIXXI
XXXXI    IIXXX   IIXXX   IIXXX   IIXXX   IIXXX   IIXXX   IXIXX   IXXIX   IXXIX
IXXII    IXXII   IXXII   IXXII   IXXII   IXXII   IXXIX   IIXXX   IXIXX   IXIXX
IIXXI    IIXXI   IIXXI   IIXXI   IIXXI   IIXXI   IXXII   IXXII   IXXII   IXXII
XXXIX    XXXIX   XXXIX   XXXIX   XXXIX   XXXXX   IIXXI   IIXXI   IIXXI   IIXXI
XXIIX    XXIIX   XXIIX   XXIIX   XXIIX   XXXIX   XXXXX   XXXXX   XXXXX   XXXXX
XXIII    XXIII   XXIII   XXIII   XXIII   XXIIX   XXXIX   XXXIX   XXXIX   XXXIX
IXXIX    IXXIX   XIXXX   XIXXX   XIXXX   XXIII   XXIIX   XXIIX   XXIIX   XXIIX
IIIXX    IIIXX   IXXIX   IXXIX   IXXIX   XIXXX   XXIII   XXIII   XXIII   XXIII
XXIXI    XXIXI   IIIXX   IIIXX   IIIXX   IXXIX   XIXXX   XIXXX   XIXXX   XIXXX
XIXXX    XIXXX   XXIXI   XXXXX   XXXXX   IIIXX   IIIXX   IIIXX   IIIXX   IIIXX
IXIXI    IXIXI   IXIXI   XXIXI   XXIXI   XXIXI   XXIXI   XXIXI   XXIXI   XXIXI
XXXXX    XXXXX   XXXXX   IXIXI   IXIXI   IXIXI   IXIXI   IXIXI   IXIXI   IXIXI
XXIXX    XXIXX   XXIXX   XXIXX   XXIXX   XXIXX   XXIXX   XXIXX   XXIXX   XXIXX
XIXII    XIXII   XIXII   XIXII   XIXII   XIXII   XIXII   XIXII   XIXII   XIXII
IIXIX    IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX
XIXIX    XIXIX   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX
XIIXX    XIIXX   XIIXX   XIIXX   XIIXX   XIIXX   XIIXX   XIIXX   XIIXX   XIIXX
XIIXI    XIIXI   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI
XIIIX    XIIIX   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX
XIIII    XIIII   XIIII   XIIII   XIIII   XIIII   XIIII   XIIII   XIIII   XIIII
IXIIX    IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX
IXIII    IXIII   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII
IIXII    IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII
IIIXI    IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI
IIIIX    IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX
IIIII    IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII
```

Figure 4.10(3) continued

```
XXXXI    XXXXI    XXXXI    XXXXI
IXXXX    IXXXX    IXXXX    IXXXX
IXXXI    IXXXI    IXXXI    IXXXI
IIXXX    IIXXX    IIXXX    IIXXX
XXXII    XXXII    XXXII    XXXII
XIXXI    XIXXI    XIXXI    XIXXI
IXXIX    XXIIX    XXIIX    XXIIX
IXIXX    IXXIX    IXXIX    IXXIX
IXXII    IXIXX    IXIXX    IXIXX
IIXXI    IXXII    IXXII    IXXII
XXXXX    IIXXI    IIXXI    IIXXI
XXXIX    XXXXX    XXXXX    XXXXX
XXIIX    XXXIX    XXXIX    XXXIX
XXIII    XXIII    XXIXI    XXIXI
XIXXX    XIXXX    XXIII    XXIII
IIIXX    IIIXX    XIXXX    XIXXX
XXIXI    XXIXI    IIIXX    IIIXX
IXIXI    IXIXI    IXIXI    IXIXI
XXIXX    XXIXX    XXIXX    XXIXX
XIXII    XIXII    XIXII    XIXII
IIXIX    IIXIX    IIXIX    IIXIX
XIXIX    XIXIX    XIXIX    XIXIX
XIIXX    XIIXX    XIIXX    XIIXX
XIIXI    XIIXI    XIIXI    XIIXI
XIIIX    XIIIX    XIIIX    XIIIX
XIIII    XIIII    XIIII    XIIII
IXIIX    IXIIX    IXIIX    IXIIX
IXIII    IXIII    IXIII    IXIII
IIXII    IIXII    IIXII    IIXII
IIIXI    IIIXI    IIIXI    IIIXI
IIIIX    IIIIX    IIIIX    IIIIX
IIIII    IIIII    IIIII    IIIII
```

Figure 4.10(3) continued

11XX1	11XX1	11XX1	11XXX	11XXX	11XXX	1XX11	1XX11	1XX11	11XXX
XX111	XX111	XX111	11XX1	1XX11	1XX11	11XXX	11XXX	11XXX	1XX11
1XX11	X1XX1	X1XX1	XX111	11XX1	11XX1	11XX1	11XX1	11XX1	11XX1
1X1X1	1XX11	1XXX1	X1XX1	XX111	XX111	XX111	XX111	XX111	XX111
XXXXX	1X1X1	1XX1X	X1XXX	X1XX1	X1XX1	X1XX1	111XX	111XX	111XX
XXXX1	XXXXX	1XX11	1XXX1	X1XXX	X1XXX	1X1X1	X1XX1	X1XX1	X1XX1
XXX1X	XXXX1	1X1X1	1XX1X	1XXX1	1XXX1	X1XXX	1X1X1	1X1X1	1X1X1
XXX11	XXX1X	XXXXX	1XX11	1XX1X	1XX1X	1XXX1	X1XXX	XXXXX	XXXXX
XX1XX	XXX11	XXXX1	1X1X1	1X1X1	1X1X1	1XX1X	1XXX1	XXX1X	XXX1X
XX1X1	XX1XX	XXX1X	111XX	111XX	11X1X	11X1X	1XX1X	X1XXX	X1XXX
XX11X	XX1X1	XXX11	XXXXX	XXXXX	111XX	111XX	11X1X	1XXX1	1XXX1
X1XXX	XX11X	XX1XX	XXXX1	XXXX1	XXXXX	XXXXX	XXXXX	1XX1X	1XX1X
X1XX1	X1XXX	XX1X1	XXX1X	XXX1X	XXXX1	XXXX1	XXXX1	11X1X	11X1X
X1X1X	X1X1X	XX11X	XXX11	XXX11	XXX1X	XXX1X	XXX1X	XXXXX	XXXXX
X1X11	X1X11	X1XXX	XX1XX	XX1XX	XXX11	XXX11	XXX11	XXX11	XXX11
X11XX	X11XX	X1X1X	XX1X1	XX1X1	XX1XX	XX1XX	XX1XX	XX1XX	XX1XX
X11X1	X11X1	X1X11	XX11X	XX11X	XX1X1	XX1X1	XX1X1	XX1X1	XX1X1
X111X	X111X	X11XX	X1X1X	X1X1X	XX11X	XX11X	XX11X	XX11X	XX11X
X1111	X1111	X11X1	X1X11	X1X11	X1X1X	X1X1X	X1X1X	X1X1X	X1X1X
1XXXX	1XXXX	X111X	X11XX	X11XX	X1X11	X1X11	X1X11	X1X11	X1X11
1XXX1	1XXX1	X1111	X11X1	X11X1	X11XX	X11XX	X11XX	X11XX	X11XX
1XX1X	1XX1X	1XXXX	X111X	X111X	X11X1	X11X1	X11X1	X11X1	X11X1
1X1XX	1X1XX	1X1XX	X1111	X1111	X111X	X111X	X111X	X111X	X111X
1X11X	1X11X	1X11X	1XXXX	1XXXX	X1111	X1111	X1111	X1111	X1111
1X111	1X111	1X111	1X1XX	1X1XX	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX
11XXX	11XXX	11XXX	1X11X	1X11X	1X1XX	1X1XX	1X1XX	1X1XX	1X1XX
11X1X	11X1X	11X1X	1X111	1X111	1X11X	1X11X	1X11X	1X11X	1X11X
11X11	11X11	11X11	11X1X	11X1X	1X111	1X111	1X111	1X111	1X111
111XX	111XX	111XX	11X11	11X11	11X11	11X11	11X11	11X11	11X11
1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X
11111	11111	11111	11111	11111	11111	11111	11111	11111	11111

Figure 4.10(4) Line Rank Tables (WC)

```
11XXX  11XXX  11XXX  X1XX1  11XXX  11XXX  11XXX  11XXX  11XXX  11XXX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1XX11  1XX11  1XX11  11XXX  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
11XX1  11XX1  11XX1  1XX11  1XX11  1XX11  1XX11  X1XXX  X1XXX  X1XXX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XX111  XX111  XX111  11XX1  11XX1  11XX1  11XX1  1XX11  1XX11  1XX11
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
111XX  111XX  111XX  XX111  XX111  XX111  XX111  11XX1  11XX1  11XX1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X1XX1  X1XXX  X1XXX  111XX  111XX  111XX  111XX  XX111  XX111  XX111
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1X1X1  X1XX1  X1XX1  X1XXX  X1XXX  X1XXX  X1XXX  111XX  111XX  111XX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XXXX1  1X1X1  1X1X1  1X1X1  1X1X1  1X1X1  1X1X1  1X1X1  1X1XX  1XXX1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XXX1X  XXXX1  XXXX1  XXXX1  XXXX1  XXXX1  XXXX1  XXXXX  1X1X1  1X1XX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X1XXX  XXX1X  XXXX1  XXXX1  XXXX1  XXXX1  XXXX1  XXXX1  XXXXX  1X1X1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1XXX1  1XXX1  XXX1X  XXX1X  XXX1X  XXX1X  XXX1X  XXX1X  XXXX1  XXXXX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1XX1X  1XX1X  1XXX1  1XXX1  1XXX1  XXX11  XXX11  XXX11  XXX1X  XXXX1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
11X1X  11X1X  1XX1X  1XX1X  1XX1X  1XXX1  1XXX1  1XXX1  XXX11  XXX1X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XXXXX  XXXXX  11X1X  11X1X  11X1X  1XX1X  1XX1X  1XX1X  1XXX1  XXX11
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XXX11  XXX11  XXX11  XXX11  XXX11  11X1X  1X1XX  1X1XX  1XX1X  1XX1X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  XX1XX  11X1X  11X1X  11X1X  11X1X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XX1XX  XX1XX  XX1XX  XX1XX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
XX11X  XX11X  XX11X  XX11X  XX11X  XX11X  XX1X1  XX1X1  XX1X1  XX1X1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  XX11X  XX11X  XX11X  XX11X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X1X11  X1X11  X1X11  X1X11  X1X11  X1X11  X1X1X  X1X1X  X1X1X  X1X1X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X11XX  X11XX  X11XX  X11XX  X11XX  X11XX  X1X11  X1X11  X1X11  X1X11
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X11X1  X11X1  X11X1  X11X1  X11X1  X11X1  X11XX  X11XX  X11XX  X11XX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X111X  X111X  X111X  X111X  X111X  X111X  X11X1  X11X1  X11X1  X11X1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
X1111  X1111  X1111  X1111  X1111  X1111  X111X  X111X  X111X  X111X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  X1111  X1111  X1111  X1111
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1X1XX  1X1XX  1X1XX  1X1XX  1X1XX  1X1XX  1XXXX  1XXXX  1XXXX  1XXXX
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
11111  11111  11111  11111  11111  11111  11111  11111  11111  11111
```

Figure 4.10(4) continued

```
11XXX  11XXX  11XXX  11XXX  11XXX  11XXX  11XXX  11XXX  XXXX1  XXXX1
X1XX1  X1XXX  X1XXX  X1XXX  X1XXX  X1XXX  X1XXX  XXXX1  11XXX  11XXX
X1XXX  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1  X1XX1  X1XXX  X1XXX  X1XXX
1XX11  1XX11  XXXX1  XXXX1  XXXX1  XXXX1  XXXX1  X1XX1  X1XX1  X1XX1
11XX1  11XX1  1XX11  1XX11  1XX11  1XX11  1XX11  1XX11  1XX11  1XX11
XX111  XX111  11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  11XX1  11XX1
111XX  111XX  XX111  XX111  XX111  XX111  XX111  XX111  XX111  XX111
1XXX1  1XXX1  111XX  111XX  111XX  111XX  111XX  111XX  111XX  111XX
1X1XX  1X1XX  1XXX1  1XXX1  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX
1X1X1  1X1X1  1X1XX  1XX1X  1XXX1  1XXX1  XXX1X  XXX1X  XXX1X  XXX1X
XXXXX  XXXXX  1X1X1  1X1XX  1XX1X  1XX1X  1XXX1  1XXX1  1XXX1  XXX1X
XXXX1  XXXX1  XXXXX  1X1X1  1X1XX  1X1XX  1XX1X  1XX1X  1XX1X  1XXX1
XXX1X  XXX1X  XXX1X  XXXXX  1X1X1  1X1X1  1X1XX  1X1XX  1X1XX  1XX1X
XXX11  XXX11  XXX11  XXX1X  XXX1X  XXX1X  1X1X1  1X1X1  1X1X1  1X1XX
1XX1X  1XX1X  XX1X1  XXX11  XXX11  XXX11  XXX11  XXX11  XXX11  1X1X1
11X1X  11X1X  1XX1X  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XX1X1  XXX11
XX1XX  XX1XX  11X1X  11X1X  11X1X  1XXXX  1XXXX  1XXXX  1XXXX  X1X11
XX1X1  XX1X1  XX1XX  XX1XX  XX1XX  11X1X  11X1X  11X1X  11X1X  1XXXX
XX11X  XX11X  XX11X  XX11X  XX11X  XX1XX  XX1XX  XX1XX  XX1XX  11X1X
X1X1X  X1X1X  X1X1X  X1X1X  X1X1X  XX11X  XX11X  XX11X  XX11X  XX1XX
X1X11  X1X11  X1X11  X1X11  X1X11  X1X1X  X1X1X  X1X1X  X1X1X  XX11X
X11XX  X11XX  X11XX  X11XX  X11XX  X1X11  X1X11  X1X11  X1X11  X1X1X
X11X1  X11X1  X11X1  X11X1  X11X1  X11XX  X11XX  X11XX  X11XX  X11XX
X111X  X111X  X111X  X111X  X111X  X11X1  X11X1  X11X1  X11X1  X11X1
X1111  X1111  X1111  X1111  X1111  X111X  X111X  X111X  X111X  X111X
1XXXX  1XXXX  1XXXX  1XXXX  1XXXX  X1111  X1111  X1111  X1111  X1111
1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X  1X11X
1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111  1X111
11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11  11X11
111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1  111X1
1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X  1111X
11111  11111  11111  11111  11111  11111  11111  11111  11111  11111
```

Figure 4.10(4) continued

```
XXXXI   XXXXI   XXXXI   XXXXI   XXXXI   XXXXI
-----   -----   -----   -----   -----   -----
IIXXX   IIXXX   IIXXX   IIXXX   IIXXX   IIXXX
-----   -----   -----   -----   -----   -----
XIXXX   XIXXX   XIXXX   XIXXX   XIXXX   XIXXX
XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI
-----   -----   -----   -----   -----   -----
IXXII   IXXII   IXXIX   IXXIX   IXXIX   IXXIX
IIXXI   IIXXI   -----   -----   -----   -----
-----   -----   IXXII   IXXII   IXXII   IXXII
XXIII   XXIII   IIXXI   IIXXI   IIXXI   IXIXX
IIIXX   IXXIX   -----   -----   -----   IIXXI
-----   IIIXX   XXIII   XXXIX   XXXIX   -----
XXXXX   -----   IIIXX   XXIII   XXIII   XXXIX
XXXIX   XXXXX   -----   IIIXX   IXIXX   XXIII
XXIXI   XXXIX   XXXXX   -----   IIIXX   IIIXX
IXXXI   XXIXI   XXXIX   XXXXX   -----   -----
IXXIX   IXXXI   XXIXI   XXIXI   XXXXX   XXXXX
IXIXX   IXIXX   IXXXI   IXXXI   XXIXI   XXIXI
IXIXI   IXIXI   IXIXX   IXIXX   IXXXI   IXXXI
-----   -----   IXIXI   IXIXI   IXIXI   IXIXI
XXXII   XXXII   -----   -----   -----   -----
XIXII   XIXII   XXXII   XXXII   XXXII   XXXII
IXXXX   IXXXX   XIXII   XIXII   XIXII   XIXII
IIXIX   IIXIX   IXXXX   IXXXX   IXXXX   IXXXX
-----   -----   IIXIX   IIXIX   IIXIX   IIXIX
XXIXX   XXIXX   -----   -----   -----   -----
XXIIX   XXIIX   XXIXX   XXIXX   XXIXX   XXIXX
XIXIX   XIXIX   XXIIX   XXIIX   XXIIX   XXIIX
XIIXX   XIIXX   XIXIX   XIXIX   XIXIX   XIXIX
XIIXI   XIIXI   XIIXX   XIIXX   XIIXX   XIIXX
XIIIX   XIIIX   XIIXI   XIIXI   XIIXI   XIIXI
XIIII   XIIII   XIIIX   XIIIX   XIIIX   XIIIX
IXIIX   IXIIX   XIIII   XIIII   XIIII   XIIII
IXIII   IXIII   IXIIX   IXIIX   IXIIX   IXIIX
IIXII   IIXII   IXIII   IXIII   IXIII   IXIII
IIIXI   IIIXI   IIXII   IIXII   IIXII   IIXII
IIIIX   IIIIX   IIIXI   IIIXI   IIIXI   IIIXI
IIIII   IIIII   IIIIX   IIIIX   IIIIX   IIIIX
                IIIII   IIIII   IIIII   IIIII
```

Figure 4.10(4) continued

1XX11	1XX11	1XX11	1XX11	1XX11	1XXX1	1XXX1	1XXX1	1XXX1	X1XXX
11XX1	11XX1	11XX1	11XX1	11XX1	1XX11	X1XX1	X1XX1	X1XXX	1XXX1
111XX	XX111	XX111	XX111	XX111	11XX1	1XX11	1XX11	X1XX1	X1XX1
XX111	111XX	111XX	111XX	X1XX1	XX111	11XX1	11XX1	1XX11	1XX11
1X1X1	1X1X1	X1XX1	X1XX1	111XX	X1XX1	XX111	XX111	11XX1	11XX1
XXXXX	XXXXX	1X1X1	X11XX	X11XX	111XX	111XX	X1XXX	XX111	XX111
XXXX1	XXXX1	XXXXX	1X1X1	1XXX1	X11XX	XXX11	111XX	1X1XX	1X1XX
XXX1X	XXX1X	XXXX1	XXXXX	1X1X1	1X1X1	X11XX	XXXX1	111XX	111XX
XXX11	XXX11	XXX1X	XXXX1	XXXXX	XXXXX	1X1XX	XXX11	XXXX1	XXXX1
XX1XX	XX1XX	XXX11	XXX1X	XXXX1	XXXX1	1X1X1	X11XX	XXX11	XXX11
XX1X1	XX1X1	XX1XX	XXX11	XXX1X	XXX1X	XXXXX	1XXXX	X11XX	X11XX
XX11X	XX11X	XX1X1	XX1XX	XXX11	XXX11	XXXX1	1X1XX	1XXXX	1XXXX
X1XXX	X1XXX	XX11X	XX1X1	XX1XX	XX1XX	XXX1X	1X1X1	1X1X1	1X1X1
X1XX1	X1XX1	X1XXX	XX11X	XX1X1	XX1X1	XX1XX	XXXXX	XXXXX	XXXXX
X1X1X	X1X1X	X1X1X	X1XXX	XX11X	XX11X	XX1X1	XXX1X	XXX1X	XXX1X
X1X11	X1X11	X1X11	X1X1X	X1XXX	X1XXX	XX11X	XX1XX	XX1XX	XX1XX
X11XX	X11XX	X11XX	X1X11	X1X1X	X1X1X	X1XXX	XX1X1	XX1X1	XX1X1
X11X1	X11X1	X11X1	X11X1	X1X11	X1X11	X1X1X	XX11X	XX11X	XX11X
X111X	X111X	X111X	X111X	X11X1	X11X1	X1X11	X1X1X	X1X1X	X1X1X
X1111	X1111	X1111	X1111	X111X	X111X	X11X1	X1X11	X1X11	X1X11
1XXXX	1XXXX	1XXXX	1XXXX	X1111	X1111	X111X	X11X1	X11X1	X11X1
1XXX1	1XXX1	1XXX1	1XXX1	1XXXX	1XXXX	X1111	X111X	X111X	X111X
1XX1X	1XX1X	1XX1X	1XX1X	1XX1X	1XX1X	1XXXX	X1111	X1111	X1111
1X1XX	1X1XX	1X1XX	1X1XX	1X1XX	1X1XX	1XX1X	1XX1X	1XX1X	1XX1X
1X11X	1X11X	1X11X	1X11X	1X11X	1X11X	1X11X	1X11X	1X11X	1X11X
1X111	1X111	1X111	1X111	1X111	1X111	1X111	1X111	1X111	1X111
11XXX	11XXX	11XXX	11XXX	11XXX	11XXX	11XXX	11XXX	11XXX	11XXX
11X1X	11X1X	11X1X	11X1X	11X1X	11X1X	11X1X	11X1X	11X1X	11X1X
11X11	11X11	11X11	11X11	11X11	11X11	11X11	11X11	11X11	11X11
111X1	111X1	111X1	111X1	111X1	111X1	111X1	111X1	111X1	111X1
1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X	1111X
11111	11111	11111	11111	11111	11111	11111	11111	11111	11111

Figure 4.10(5) Line Rank Tables (W4)

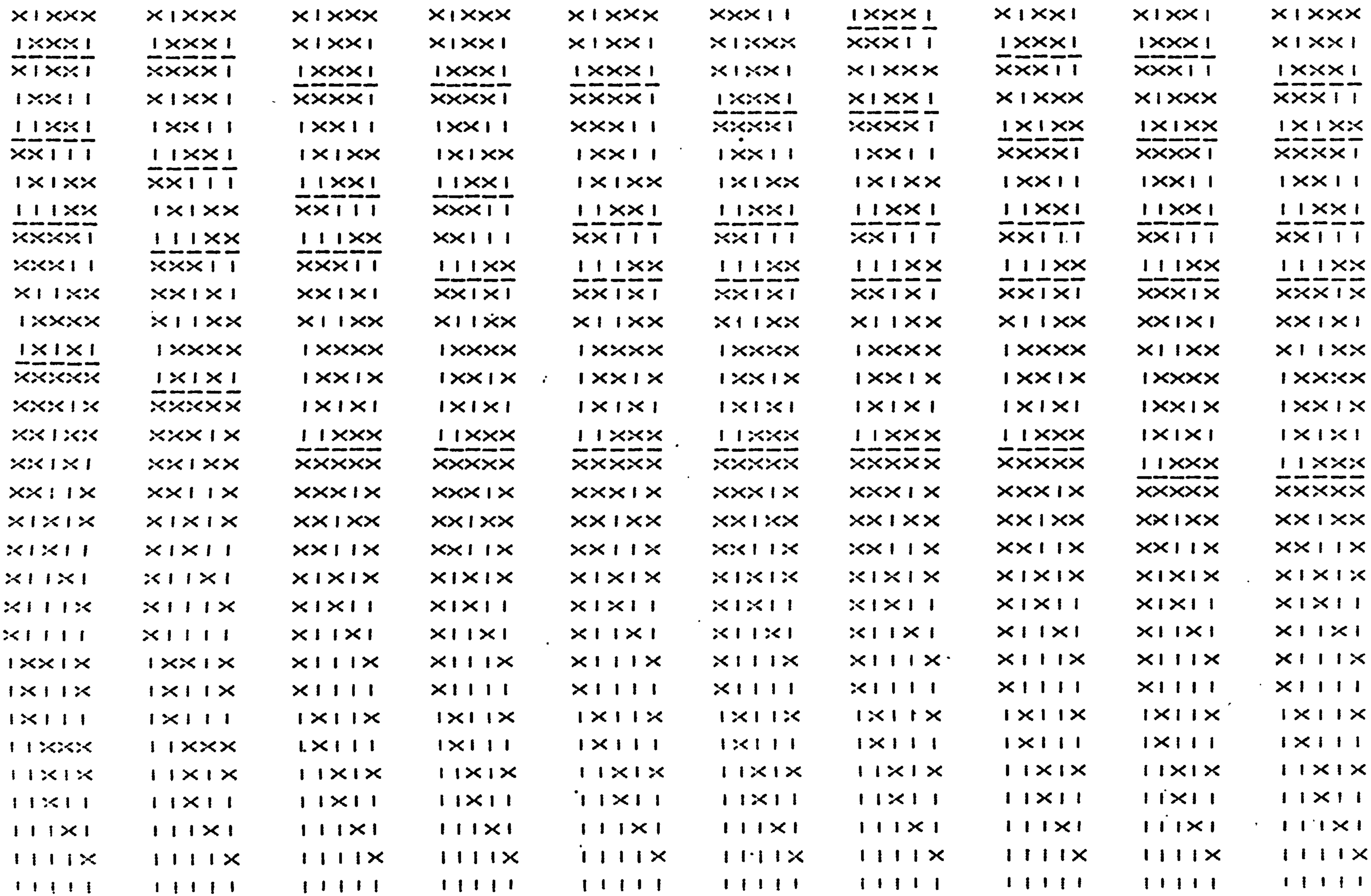


Figure 4.10(5) continued

```
XIXXX   XXXII   XXXII   XXXII   XXXII   XXXII   XXXII   XXXII   XXXII   XXXII
XIXXI   XIXXX   XIXXX   XIXXX   XIXXX   XIXXX   XIXXX   XIXXX   XIXXX   XIXXX
IXXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI   XIXXI
XXXII   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI   IXXXI
IXIXX   IXIXX   IXIXX   IXIXX   IXIXX   IXIXX   IXIXX   IXXII   IXIXX   IXIXX
XXXXI   XXXXI   XXXXI   XXXXI   XXXXI   XXXXI   IIXXI   IXIXX   IXXII   XXXXI
IXXII   IXXII   IXXII   IXXII   IXXII   IXXII   XXXXI   IIXXI   IIXXI   IXXII
IIXXI   IIXXI   IIXXI   IIXXI   IIXXI   IIXXI   XXIII   XXXXI   XXXXI   IIXXI
XXIII   XXIII   XXXIX   XXXIX   XXXIX   XXXIX   IXXII   XXIII   XXIII   XXIII
IIIXX   IIIXX   XXIII   XXIII   XXIXI   XXIXI   IIIXX   IIIXX   IIIXX   IIIXX
XXXIX   XXXIX   IIIXX   IIIXX   XXIII   XXIII   XXXIX   XXXIX   XXXIX   XXXIX
XXIXI   XXIXI   XXIXI   XXIXI   IXIXI   IXXIX   XXIXI   XXIXI   XXIXI   XXIXI
XIIXX   XIIXX   XIIXX   XIXII   IIIXX   IXIXI   IXXIX   IXXIX   IXXIX   IXXIX
IXXXX   IXXXX   IXXXX   XIIXX   XIXII   IIIXX   IXIXI   IXIXI   IXIXI   IXIXI
IXXIX   IXXIX   IXXIX   IXXXX   XIIXX   XIXII   XIXII   XIXII   XIXII   XIXII
IXIXI   IXIXI   IXIXI   IXXIX   IXXXX   XIIXX   XIIXX   XIIXX   XIIXX   XIIXX
IIXXX   IIXXX   IIXXX   IXIXI   IXXIX   IXXXX   IXXXX   IXXXX   IXXXX   IXXXX
XXXXX   XXXXX   XXXXX   IIXXX   IIXXX   IIXXX   IIXXX   IIXXX   IIXXX   IIXXX
XXIXX   XXIXX   XXIXX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX
XXIII   XXIII   XXIII   XXXXX   XXXXX   XXXXX   XXXXX   XXXXX   XXXXX   XXXXX
XIXIX   XIXIX   XIXIX   XXIXX   XXIXX   XXIXX   XXIXX   XXIXX   XXIXX   XXIXX
XIXII   XIXII   XIXII   XXIIX   XXIIX   XXIIX   XXIIX   XXIIX   XXIIX   XXIIX
XIIXI   XIIXI   XIIXI   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX
XIIIX   XIIIX   XIIIX   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI
XIIII   XIIII   XIIII   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX
IXIIX   IXIIX   IXIIX   XIIII   XIIII   XIIII   XIIII   XIIII   XIIII   XIIII
IXIII   IXIII   IXIII   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX
IIXIX   IIXIX   IIXIX   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII
IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII
IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI
IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX
IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII
```

Figure 4.10(5) continued

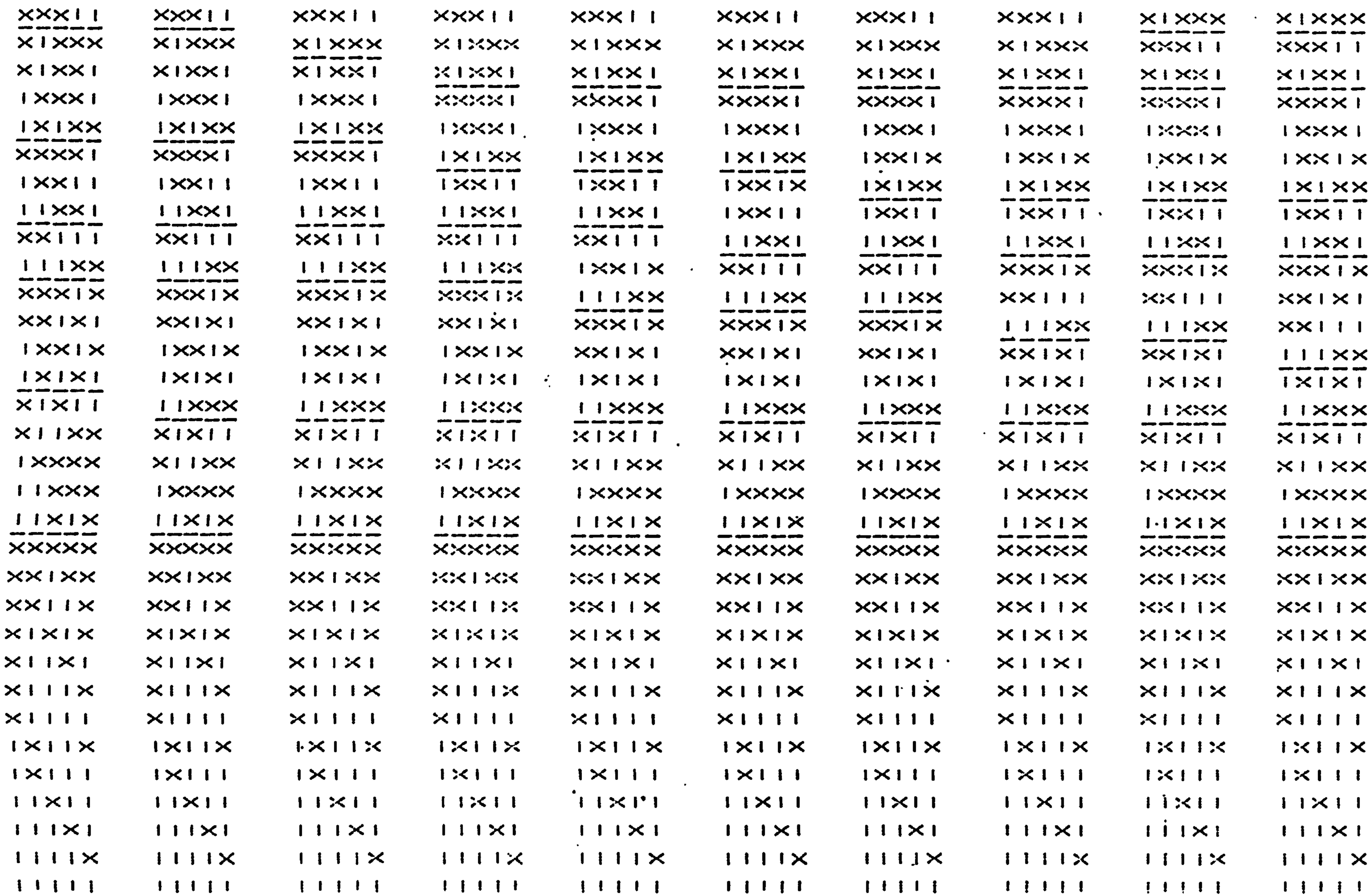


Figure 4.10(5) continued

```
XIXXX  XIXXX  XIXXX  XIXXX  XIXXX  XXXXI  XXXXI  XXXXI  XXXXI  XXXXI
XXXII  XXXII  XXXII  XXXXI  XXXXI  XIXXX  XIXXX  XIXXX  XIXXX  XIXXX
XIXXI  XIXXI  XIXXI  XXXII  XXXII  IXXXI  IXXXI  IXXXI  IXXXI  IXXXI
XXXXI  XXXXI  XXXXI  XIXXI  XIXXI  XXXII  XXXII  XXXII  XXXII  XXXII
IXXXI  IXXXI  IXXXI  IXXXI  IXXXI  XIXXI  XIXXI  XIXXI  XIXXI  XIXXI
IXXIX  IXXIX  IXXIX  IXXIX  IXXIX  IXXIX  XXIXI  IXXIX  IXXIX  IXXIX
IXIXX  IXIXX  IXIXX  IXIXX  IXIXX  IXIXX  IXXIX  XXIXI  XXIXI  XXIXI
IXXII  XXIXI  XXIXI  IIXXX  IIXXX  IIXXX  IXIXX  IXIXX  IXIXX  IXIXX
IIXXI  IXXII  IXXII  XXIXI  XXIXI  XXIXI  IIXXX  IIXXX  IIXXX  IIXXX
XXXIX  IIXXI  IIXXX  IXXII  IXXII  IXXII  IXXII  IXXII  XXXIX  XXXIX
XXIXI  XXXIX  IIXXI  IIXXI  IIXXI  IIXXI  IIXXI  IIXXI  IXXII  IXXII
XXIII  XXIII  XXXIX  XXXIX  XXXIX  XXXIX  XXXIX  XXXIX  IIXXI  IIXXI
IIIXX  IIXXX  XXIII  XXIII  XXIII  XXIII  XXIII  XXIII  XXIII  XXIII
IXIXI  IIIXX  IIIXX  IIIXX  IIIXX  IIIXX  IIIXX  IIIXX  IIIXX  IIIXX
IIXXX  IXIXI  IXIXI  IXIXI  IXIXI  IXIXI  IXIXI  IXIXI  IXIXI  IXXXX
XIXII  XIXII  XIXII  XIXII  XIXII  XIXII  XIXII  XIXII  XIXII  IXIXI
XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIIXX  XIXII
IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  IXXXX  XIIXX
IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX  IIXIX
XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX
XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX  XXIXX
XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX  XXIIX
XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX  XIXIX
XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI  XIIXI
XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX  XIIIX
XIIII  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII  XIIII
IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX  IXIIX
IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII  IXIII
IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII  IIXII
IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI  IIIXI
IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX  IIIIX
IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII  IIIII
```

Figure 4.10(5) continued

```
XXXXI
XIXXX
IXXXI
XXXII
XIXXI
IXXIX
XXIXI
IXIXX
IIXXX
XXXIX
IXXII
IIXXI
XXIII
IIIXX
IXXXX
IXIXI
XIXII
XIIXX
IIXIX
XXXXX
XXIXX
XXIII X
XIXIX
XIIXI
XIIIX
XIIII
IXIIX
IXIII
IIXII
IIIXI
IIIIX
IIIII
```

Figure 4.10(5) continued

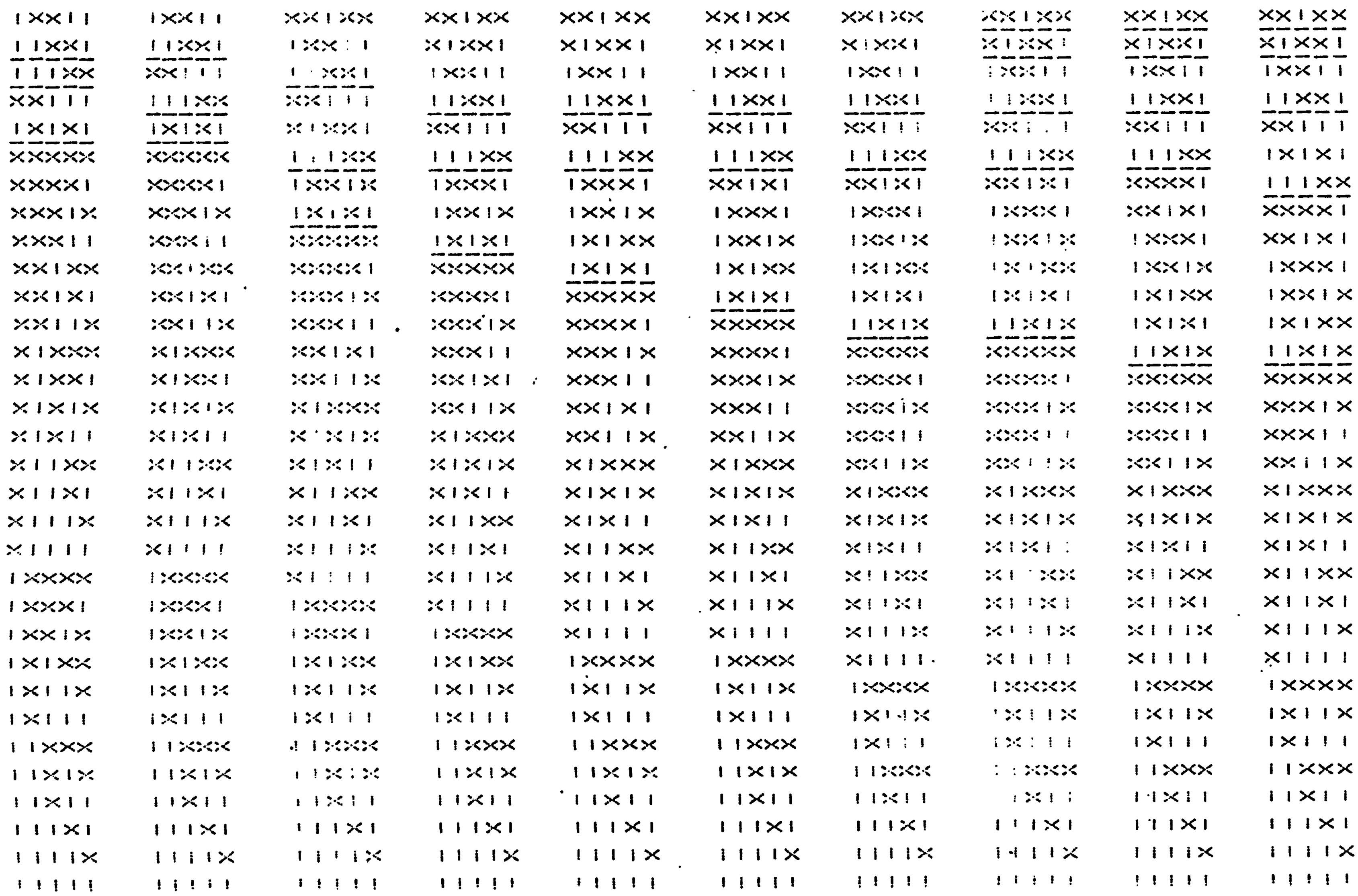


Figure 4.10(6) Line Rank Tables (WE)

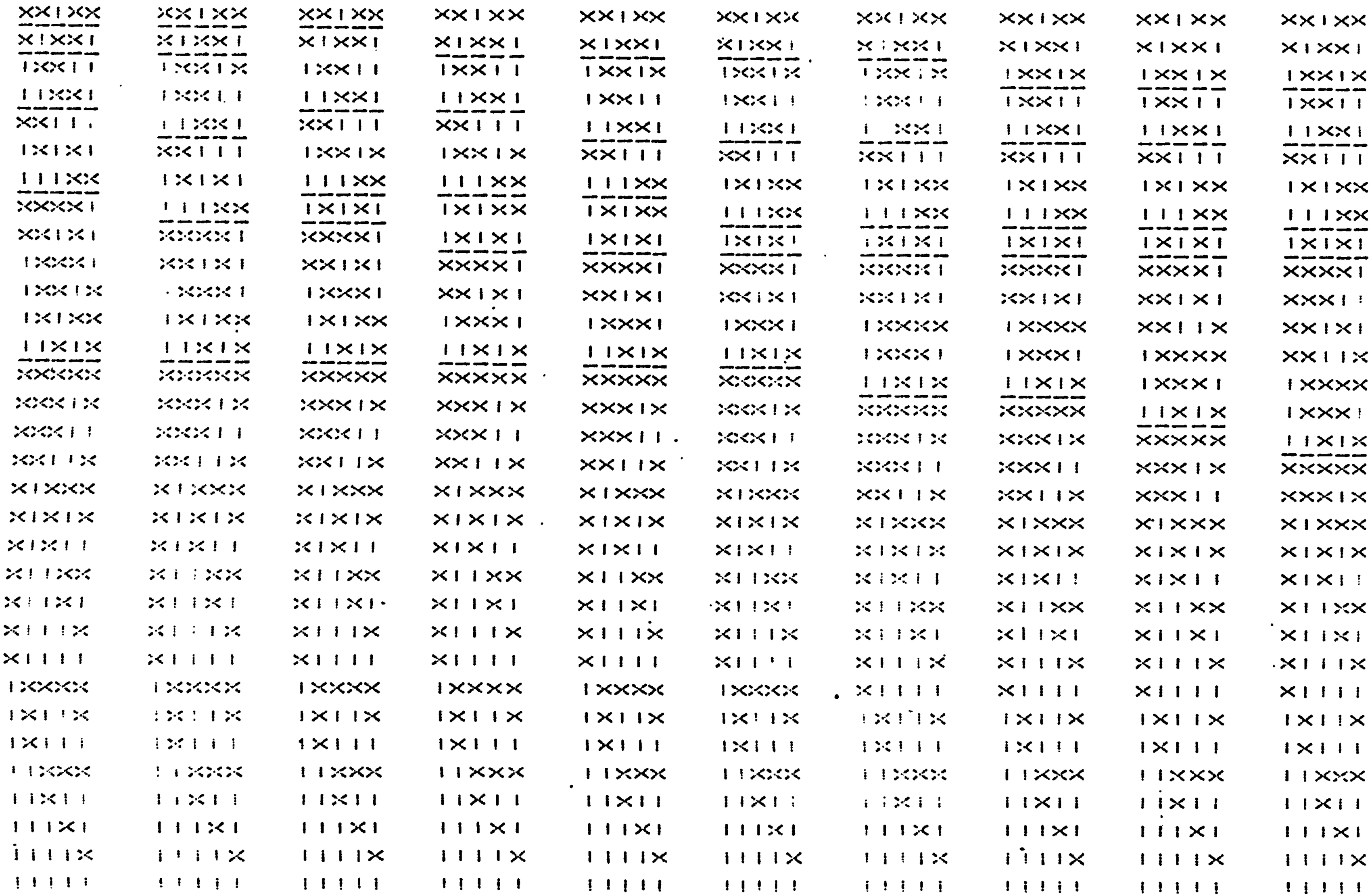


Figure 4.10(6) continued

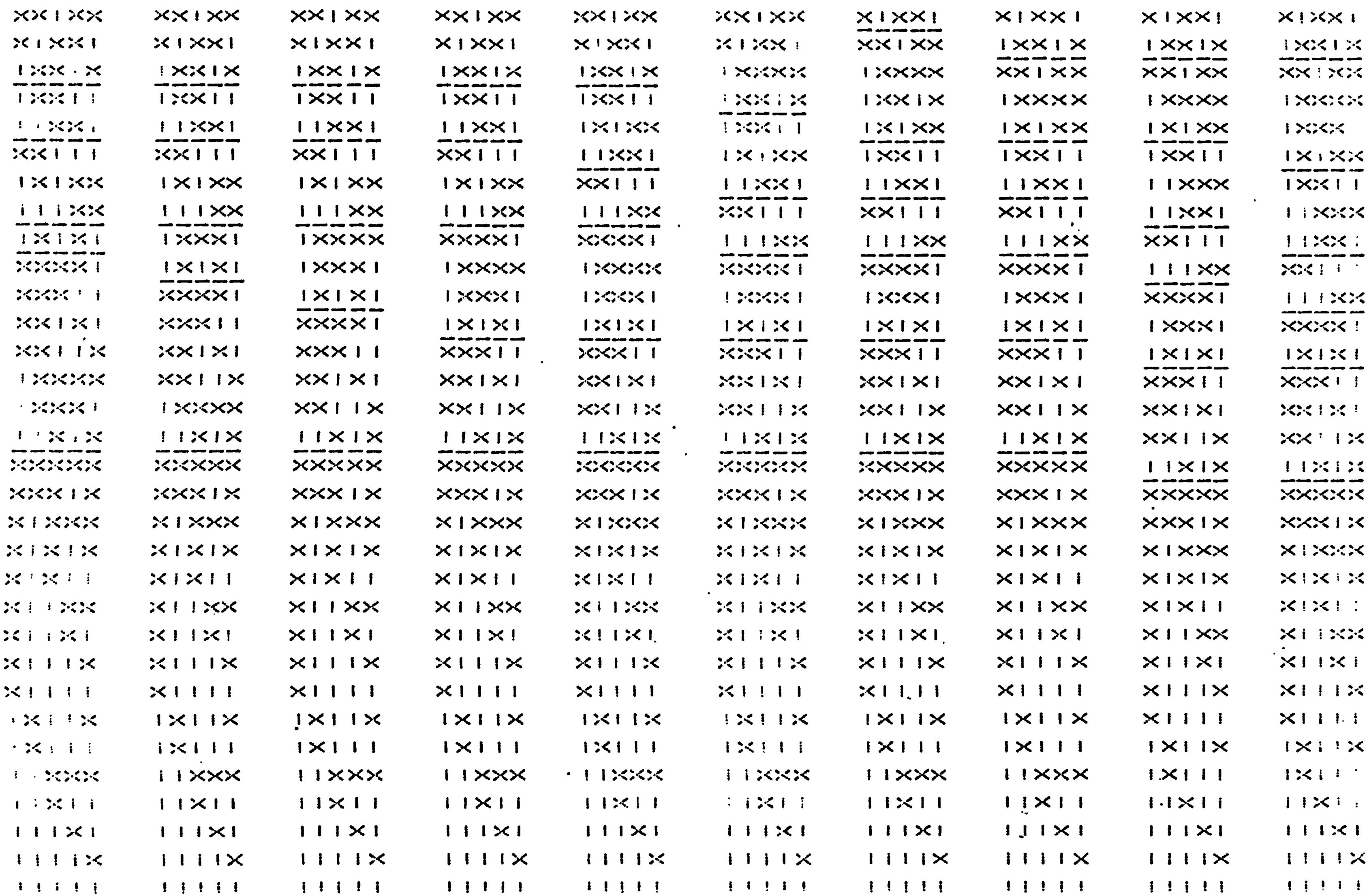


Figure 4.10(6) continued

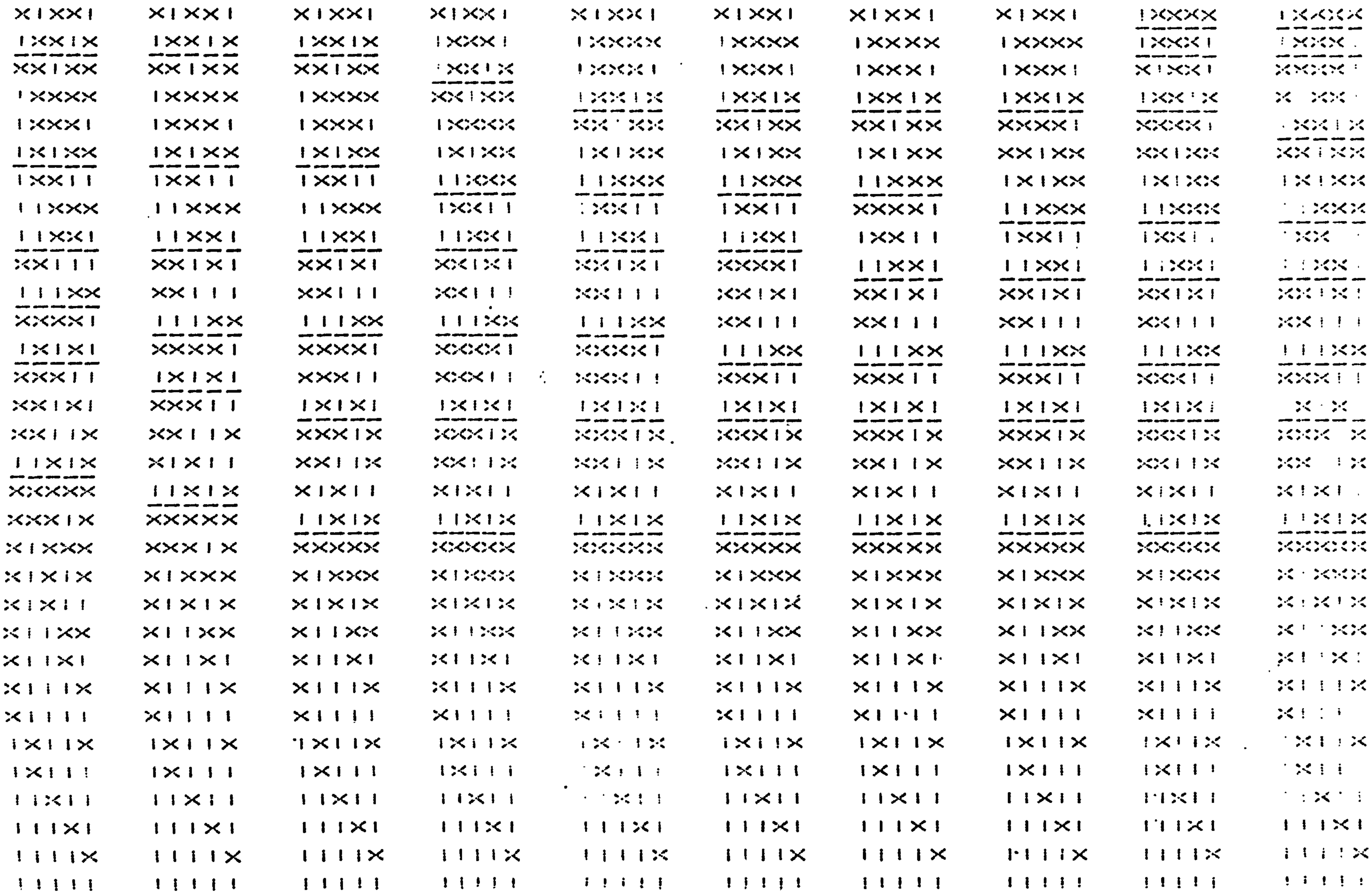


Figure 4.10(6) continued

```
IXXXX   IXXXX   IXXXX   IXXXX   IXXXX   XXXXI   XXXXI
IXXXI   IXXXI   IXXXI   IXXXI   XXXXI   IXXXX   IXXXX
XXXXI   XXXXI   XXXXI   XXXXI   IXXXI   IXXXI   IXXXI
XIXXI   XIXXI   XIXXI   XXXII   XXXII   XXXII   XXXII
IXXIX   IXXIX   IXXIX   XIXXI   XIXXI   XIXXI   XIXXI
XXIXX   XXXII   IIXXX   IXXIX   IXXIX   IXXIX   IXXIX
IXIXX   XXIXX   XXXII   IIXXX   IIXXX   IIXXX   IIXXX
IIXXX   IXIXX   XXIXX   XXIXX   XXIXX   XXIXX   XXIXX
IXXII   IIXXX   IXIXX   IXIXX   IXIXX   IXIXX   XXIXI
IIXXI   IXXII   IXXII   XXIXI   XXIXI   XXIXI   IXIXX
XXIXI   IIXXI   IIXXI   IXXII   IXXII   IXXII   IXXII
XXIII   XXIXI   XXIXI   IIXXI   IIXXI   IIXXI   IIXXI
IIIXX   XXIII   XXIII   XXIII   XXIII   XXIII   XXIII
XXXII   IIIXX   IIIXX   IIIXX   IIIXX   IIIXX   IIIXX
IXIXI   IXIXI   IXIXI   IXIXI   IXIXI   IXIXI   IXIXI
XXXIX   XXXIX   XXXIX   XXXIX   XXXIX   XXXIX   XXXIX
XXIIX   XXIIX   XXIIX   XXIIX   XXIIX   XXIIX   XXIIX
XIXII   XIXII   XIXII   XIXII   XIXII   XIXII   XIXII
IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX   IIXIX
XXXXX   XXXXX   XXXXX   XXXXX   XXXXX   XXXXX   XXXXX
XIXXX   XIXXX   XIXXX   XIXXX   XIXXX   XIXXX   XIXXX
XIXIX   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX   XIXIX
XIIXX   XIIXX   XIIXX   XIIXX   XIIXX   XIIXX   XIIXX
XIIXI   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI   XIIXI
XIIIX   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX   XIIIX
XIIII   XIIII   XIIII   XIIII   XIIII   XIIII   XIIII
IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX   IXIIX
IXIII   IXIII   IXIII   IXIII   IXIII   IXIII   IXIII
IIXII   IIXII   IIXII   IIXII   IIXII   IIXII   IIXII
IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI   IIIXI
IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX   IIIIX
IIIII   IIIII   IIIII   IIIII   IIIII   IIIII   IIIII
```

Figure 4.10(6) continued

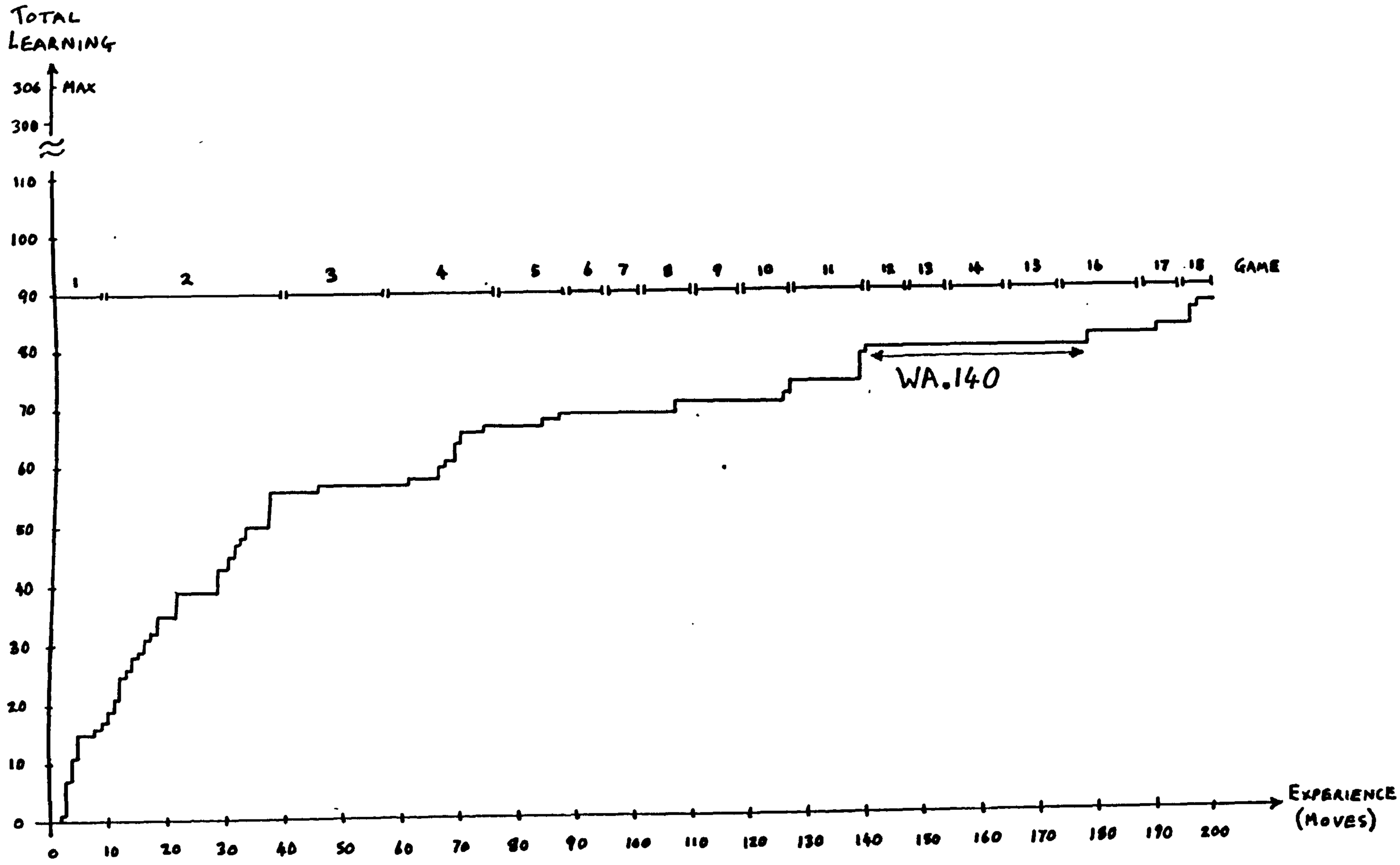


Figure 4.11(1) Total Learning Curve (WA)

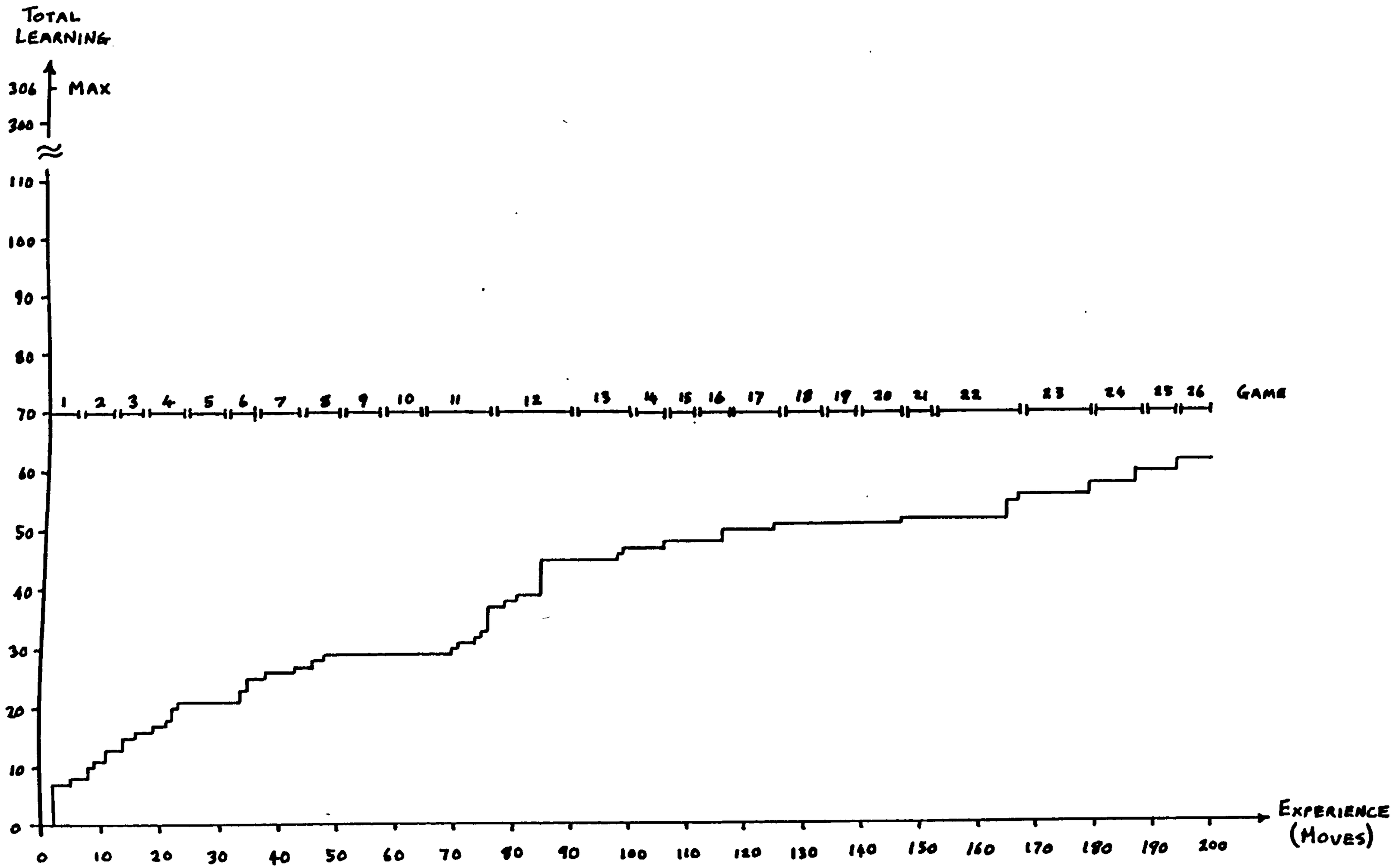


Figure 4.11(2) Total Learning Curve (W6)

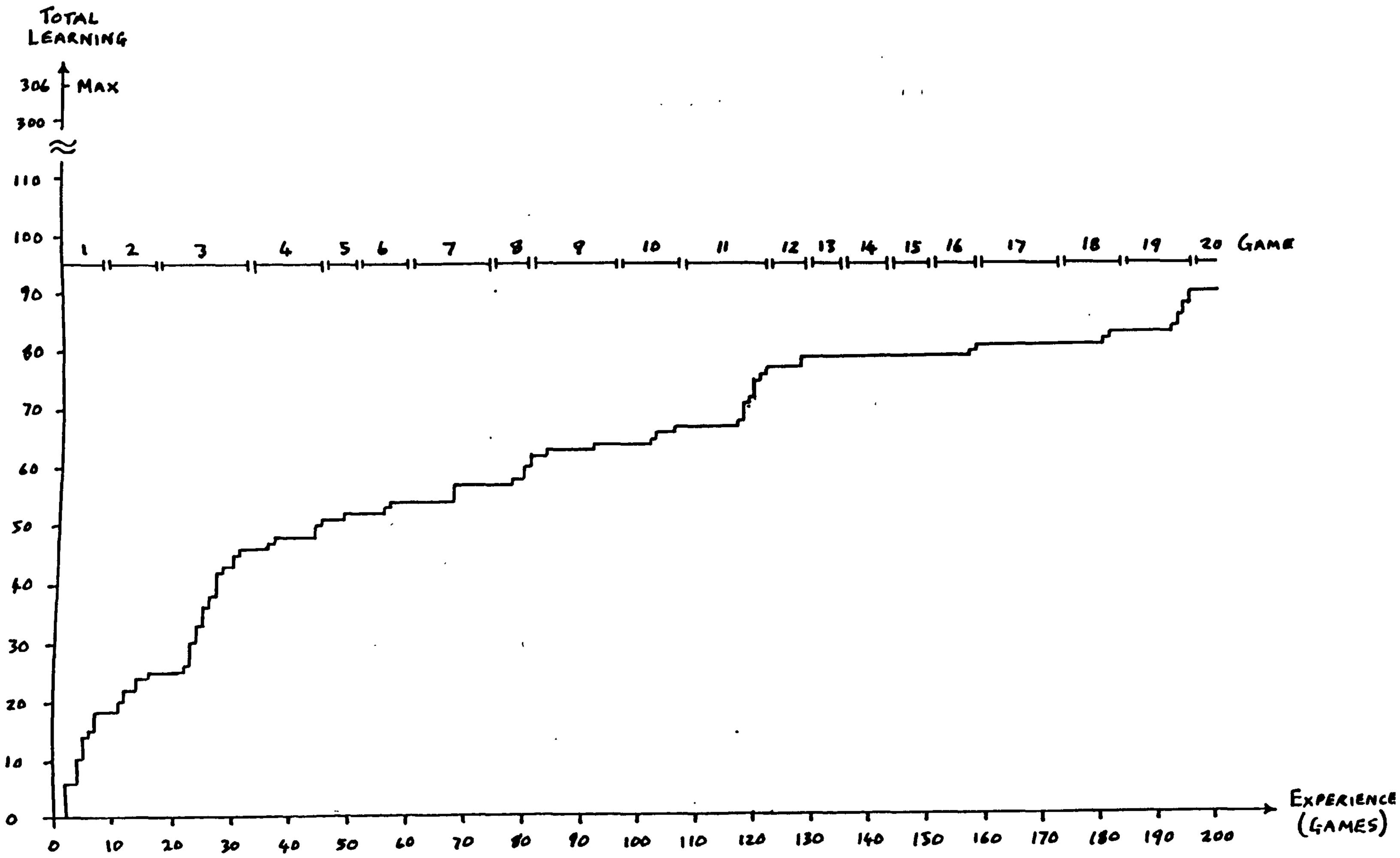


Figure 4.11(3) Total Learning Curve (W8)

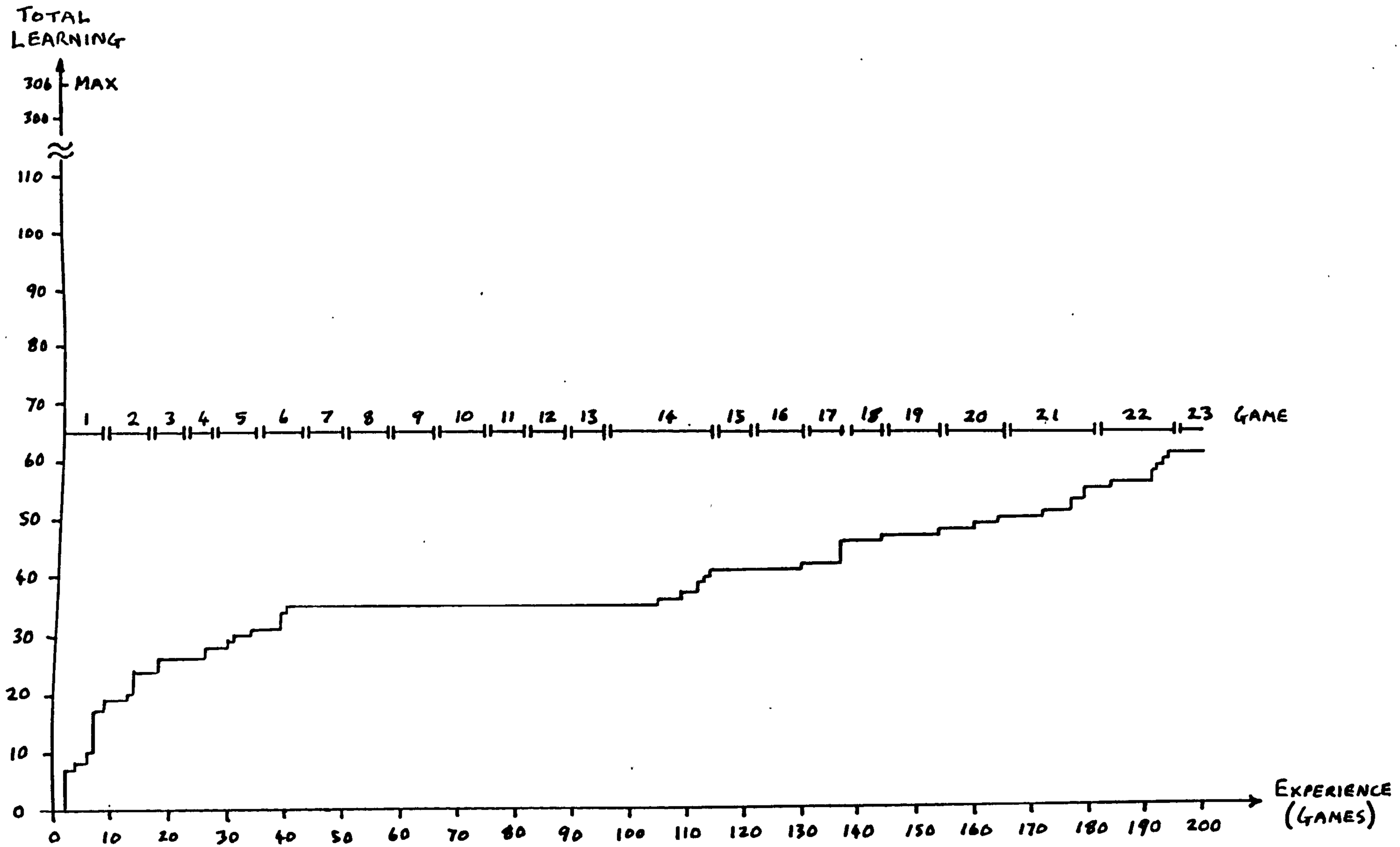


Figure 4.11(4) Total Learning Curve (WC)

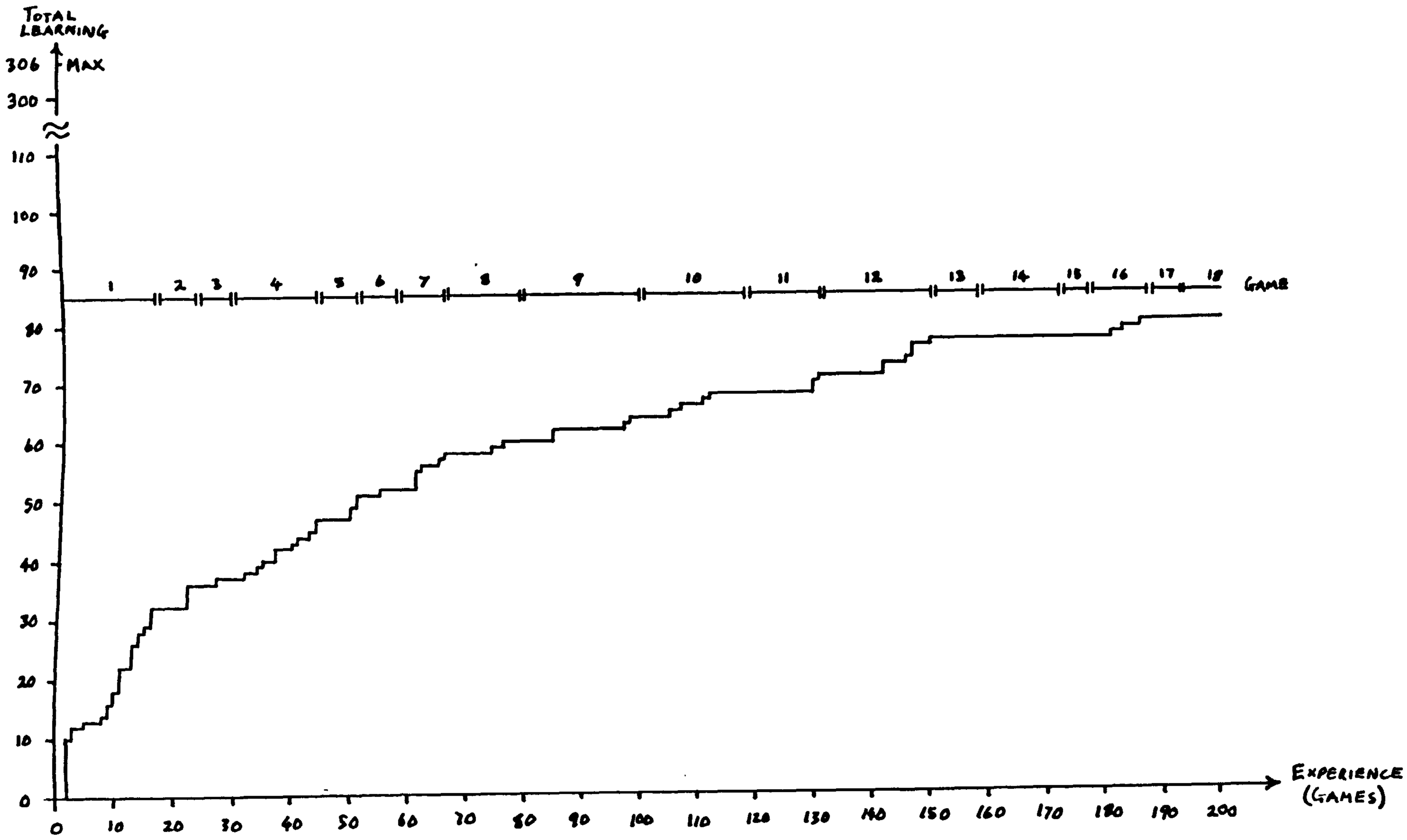


Figure 4.11(5) Total Learning Curve (W4)

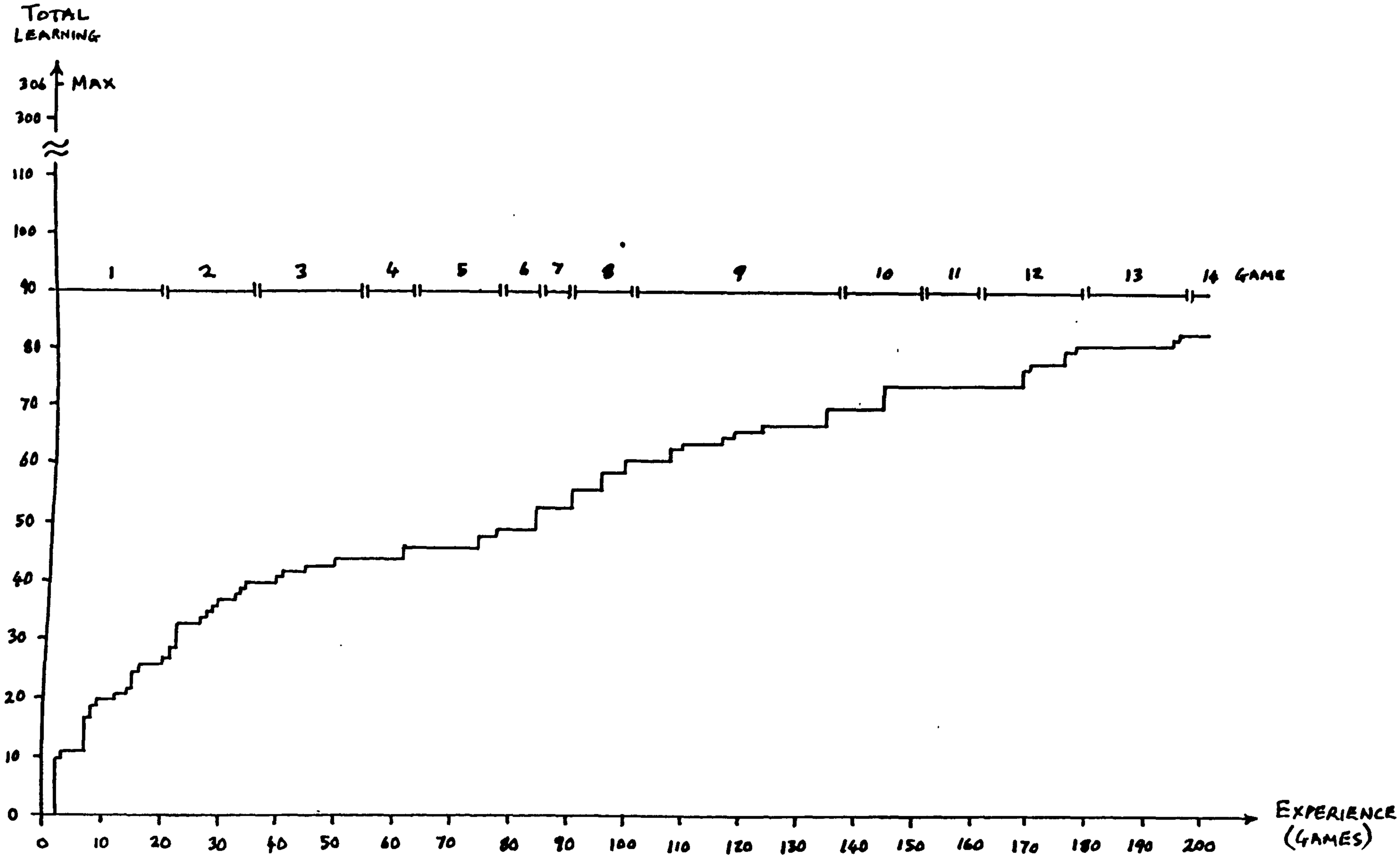


Figure 4.11(6) Total Learning Curve (WE)

4.7.2 Line Value Tabulations

Fig. 4.8(1-6) tabulates the value assigned to each line pattern by the line evaluation element after each move during training. The six sets of results (1-6) correspond to the six training schemes of fig. 4.21. Changes in the value of a line pattern (i.e. learning events) are highlighted by bold face figures. The upper tables show, for each line pattern, the actual values stored in the lookup table (line evaluation element of section 3.4.3.1). That is, high numbers imply high rank in the upper tables of fig. 4.8, while the lower tables rank the lines from the best (rank 1) down to the worst (i.e. low numbers imply high rank). The training which produced these tables is given in section 4.9.

The tables of fig. 4.8 show most activity at the beginning of learning followed by gradually more sparsely spread bursts of isolated change in the learning element. No obvious differences are seen due to differences in the training method over the six tables.

4.7.3 Condensed Line Value Tables

Fig. 4.9 tabulates the same information as fig. 4.8 (section 4.7.2) but the moves (columns) containing no changes (no bold face figures) have been removed to give a concise log of the learning. Again the tables were generated by the training to be discussed in section 4.9.

4.7.4 Line Rank Tables

Fig. 4.10(1-6) tabulates the rank after each change in the line evaluation element. Thus each column corresponds to a column of fig. 4.9 (section 4.7.3) sorted by rank, high ranking line patterns appearing

at the top. Line patterns are grouped together with other patterns of same rank. The dashed lines separates line patterns differing in rank by one.

The battle for rank between line patterns can be followed by tracing the path of a line pattern as has been done for the top ranking lines in fig. 4.10(1).

4.7.5 Total Learning Curves

The graphs in fig. 4.11(1-6) show the "total learning" in the line evaluation element (section 3.4.3.1) plotted against number of training moves encountered. Each point on the graph represents the total number of changes ("reward" increments) to elements of the evaluation element calculated as the sum, after each training move, of all line rank values in the lookup table (line evaluation element). That is, a step of n increments in the graph indicates n learning changes to the line evaluation element during the corresponding move. These n changes might all be changes of rank of the same line or of several different lines.

Each change of rank will rerank the "rewarded" line pattern one higher than previously in response to the teacher's advice that the line is better than an equally ranked line. Hence, the sum of line ranks (values) can only increase with learning or stay the same, which accounts for the absence of negative gradients in the graphs of fig. 4.11. The curves climb towards a maximum of 306, given by summing the fully trained evaluation element of fig. 4.14.

The graphs are obtained by plotting the sum of each column of the (upper) rank tables of fig. 4.8. The curves of fig. 4.11(2) and 4.11(4) are similar in shape and show a low level of learning. This similarity may be due to these training methods involving a better opponent than those used to generate the other curves.

Figs. 4.11(5) and 4.11(6) have a similar smooth shape which may derive from the two players involved in each training being of similar playing strength to their respective opponents.

No claims are made for these similarities in view of the lack of further supporting results, and the similarity of fig 4.11(1) and 4.11(3) which has no obvious cause. The total learning graphs of fig. 4.11 are discussed further in section 4.7.6.

4.7.6 Discussion of the Line Evaluation Element

The line evaluation net of section 3.4.3.1 may be seen as an approximation to an "ideal" evaluation element. An "ideal" line evaluation would give, not only the relative ranking (i.e. which is "better") of the two line patterns presented to it, but an indication of the degree to which one line is "better" than the other. This implies a lookup table analogus to a town-to-town mileage chart, except that the line evaluation table has a sign associated with the distance between lines indicating whether line A is "better" than line B or "worse". The table would be accessed as a mileage chart is, taking the value at the intersection of the line A row with the line B column as the relative advantage A has over B. The table would be updated (during learning) by incrementing or decrementing the distance between A and B on advice from the teacher.

The difficulty with this "ideal" evaluation element is that in the game a change in distance between A and B can also imply changes in the distances between the other lines and both A and B. These implied changes are not reflected in the chart since the system is not linear in that distance A to C is not necessarily the sum of distances A to B and B to C.

The simplification involved in converting the "ideal" evaluation to the rank evaluation of section 3.4.3.1 involves two steps. First, the table has been collapsed to a one dimensional rank table by summing the distances from each line to all other lines. Access to this rank table now requires an access for each of the two lines and a subtraction of the ranking values. The evaluation is further simplified by returning only the sign of the difference in rank.

The resulting ranking is now linear in that if A is better than B and B better than C, then, A is better than C. But, since only the sign of the difference has been retained, A is better than B, and C is better than B, implies nothing between A and C. More importantly, the collapse to a single column ranking table allows incrementation of the distance between A and B to be reflected also as changes in the distances (and hence possibly the sign) between A and other lines, or B and other lines, whichever of the two is incremented.

A training algorithm which increments (raises by 1) the rank of a line when it is ruled better than another line by the teacher thus gives a learning element in which the various lines in the table "compete" for ranking position by "climbing" over lesser ranking lines. By putting a further constraint of allowing a line to "climb" only when it competes with a line of equal rank ensures the learning process will eventually terminate in perfect rank order given consistant teaching.

During training the ranking may become out of order if a particular line is encountered (and reranked) more often than a better line.

We might therefore expect the "performance" of the evaluation element during training to exhibit a "sawtooth" learning curve as the predominance of a particular line causes the ranking to become out of order. The "performance" should then resume its upward trend as the active line runs out of lesser lines to "climb up" and itself becomes a stepping stone for better lines.

Unfortunately, as explained in section 4.7.1, the learning curves obtained do not clearly display this effect.

The "total learning" curves (section 4.7.5) of fig. 4.11 and tables of fig. 4.8, do however, show long stable periods during training when the learning element (line evaluation net) remains unchanged, followed by a period of rapid change stabilizing again when the active lines exhaust the supply of lesser lines to climb upon. The rapid change begins again when a new line is encountered thus supplying a lesser line to climb on, or the new line itself becomes an active climber and in turn re-activates other lines and so on until the activity again "burns itself out".

When the most often encountered lines become separated (correctly or incorrectly) in the ranking order they become isolated (no other often-encountered lines are equal in rank with them) and so a period of stability ensues. The stability is lost when the course of the game brings a new line into activity (and into the class of often-encountered lines).

We would expect the curves of fig. 4.11 to have the form of an asymptotic learning curve since stable periods become more common as the lines become separated in rank with learning and approach the ideal ranking order implied in the advice of the teacher.

Initially (before training) all the lines are ranked at the same level so reranking of the better of two equally ranked lines occurs frequently thus producing the steep rise in the initial portion of the curves of fig. 4.11.

4.8 Parameters

The parameters listed in fig. 4.12, and discussed now, arise out of the move search algorithm of sections 3.4.2 and 3.4.4 and figs. 3.12 and 4.3.

The list of parameters in System 3 (fig. 4.12) is too long to do an exhaustive evaluation of the effect of each parameter on the system as a whole. An exhaustive evaluation is made more impractical by the inter-related effects of some parameters on others. For example, the effect of the CONFIDENCE parameter on the ATTACKING and DEFENDING parameters, this effect itself being dependant on the setting of DON, DOFF, AON and AOFF as explained later in section 4.8.3.

The effects of the parameters discussed below are also intimately related to the training of the line evaluation element discussed in sections 4.6 and 4.7 above.

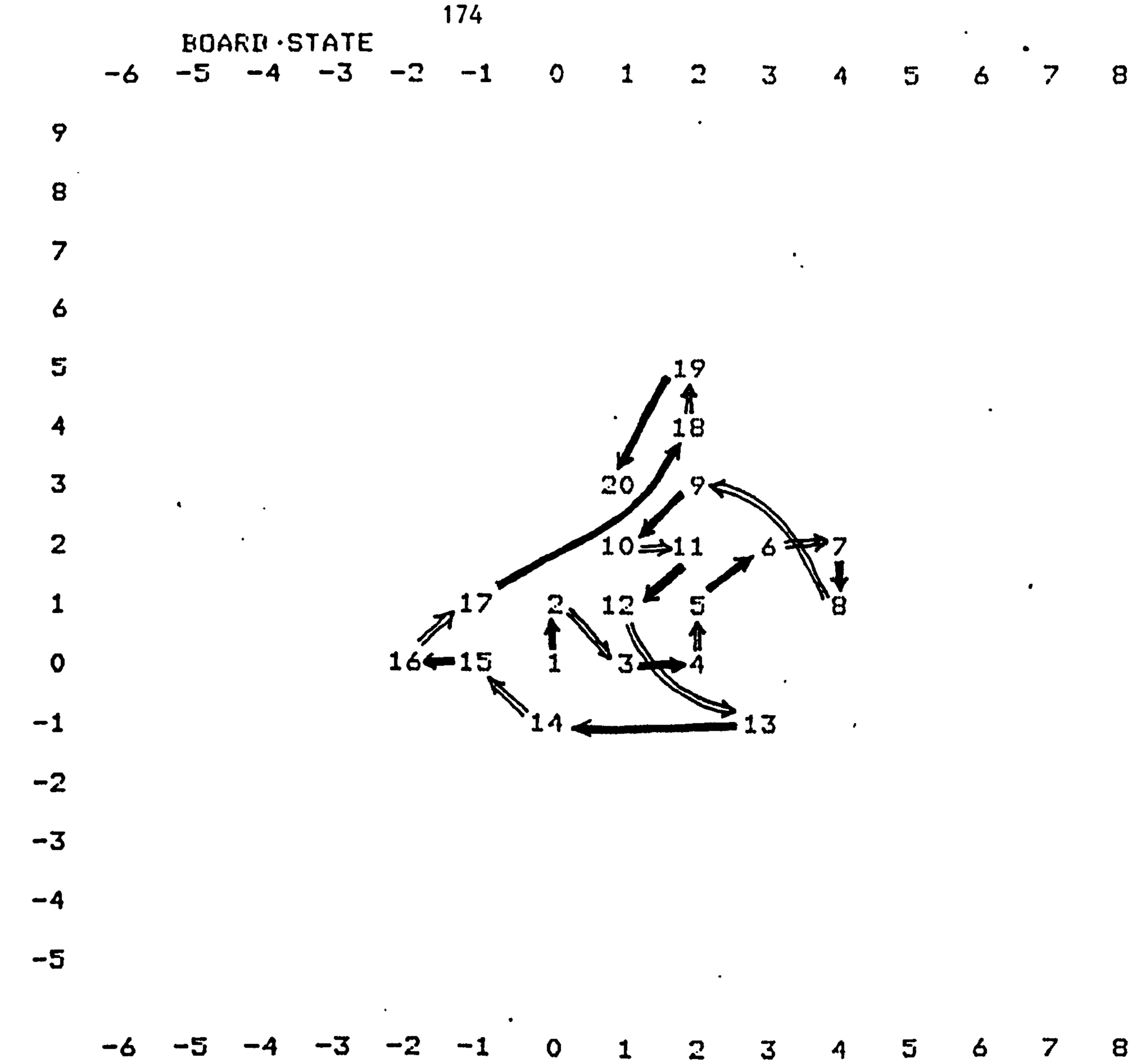
A restricted number of parameter variations are investigated here with the presentation and discussion of some results. The settings of those parameters not directly under examination will be, in most cases, no more than arbitrary, or at best "educated guesses". Unless indicated to the contrary these parameter settings will be as in fig. 4.23.

Parameter	Possible Values	Sections
ATTACKING	True/False	4.8.1 and 4.8.3
DEFENDING	True/False	4.8.1 and 4.8.3
CONFIDENCE	$-\infty..+\infty$	4.8.2 and 4.8.3
DON	$-\infty..+\infty$	4.8.3
DOFF	$-\infty..+\infty$	4.8.3
AON	$-\infty..+\infty$	4.8.3
AOFF	$-\infty..+\infty$	4.8.3
INTEREST	$0..\infty$	4.8.5
TRYING	True/False	
LEARNING	True/False	
MEDIOCRE	$0..\infty$	4.8.4
OUTSTANDING	$0..\infty$	4.8.4
MAXDEADENDS	$1..\infty$	4.8.5
NODEPATIENCE	$0..\infty$	4.8.5

Figure 4.12 Parameter Values

4.8.1 DEFENDING and ATTACKING Parameters

The DEFENDING and ATTACKING parameters used in Box 1 and Box 6 of the move search flow chart (section 3.4.4.1 and 3.4.4.6, fig. 3.12) control which lines are investigated during a move search. The effect of these parameters on the sequence of moves during a game is illustrated in fig. 4.13.



ODD MOVES ARE WHITE

*DG

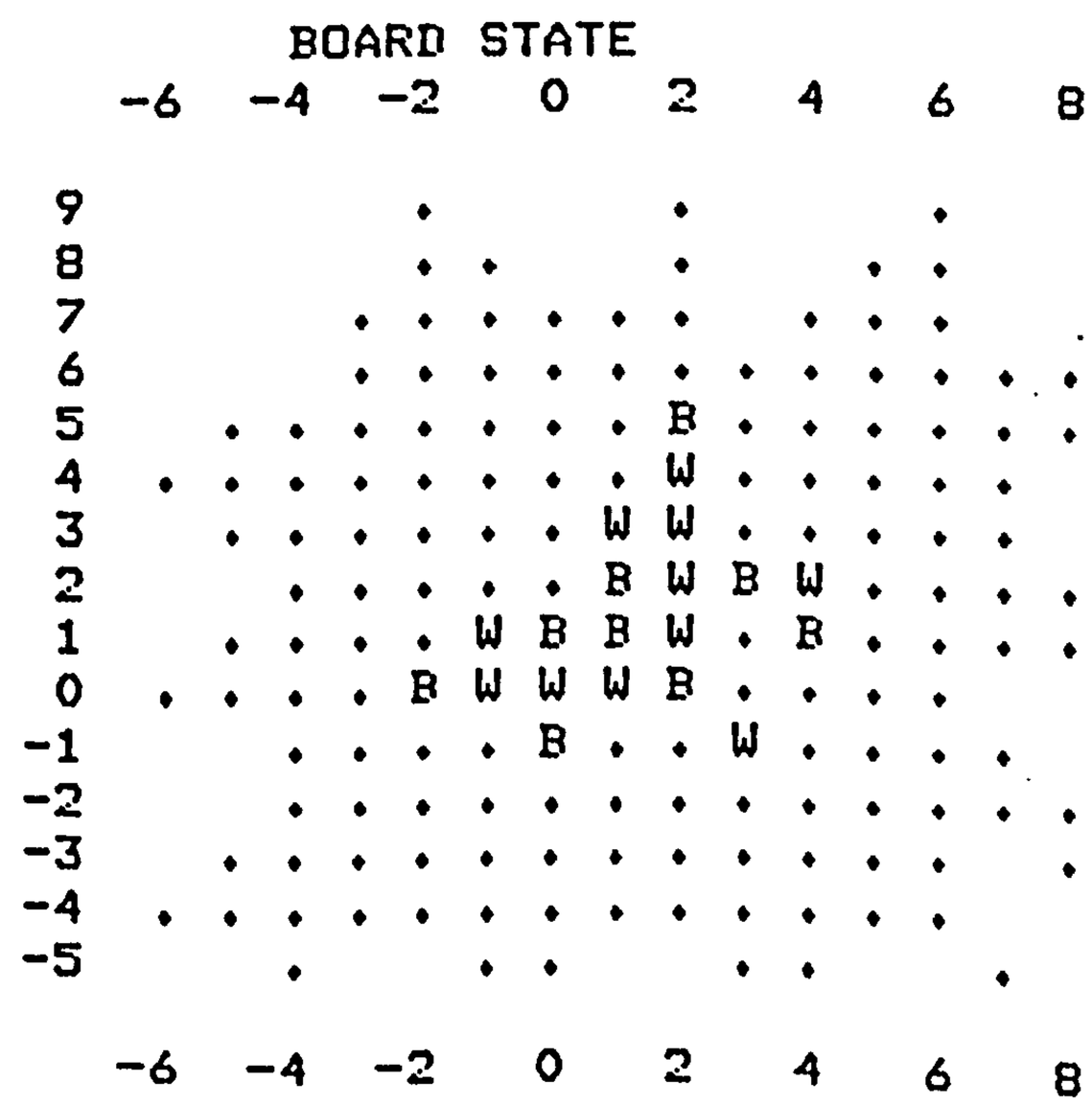
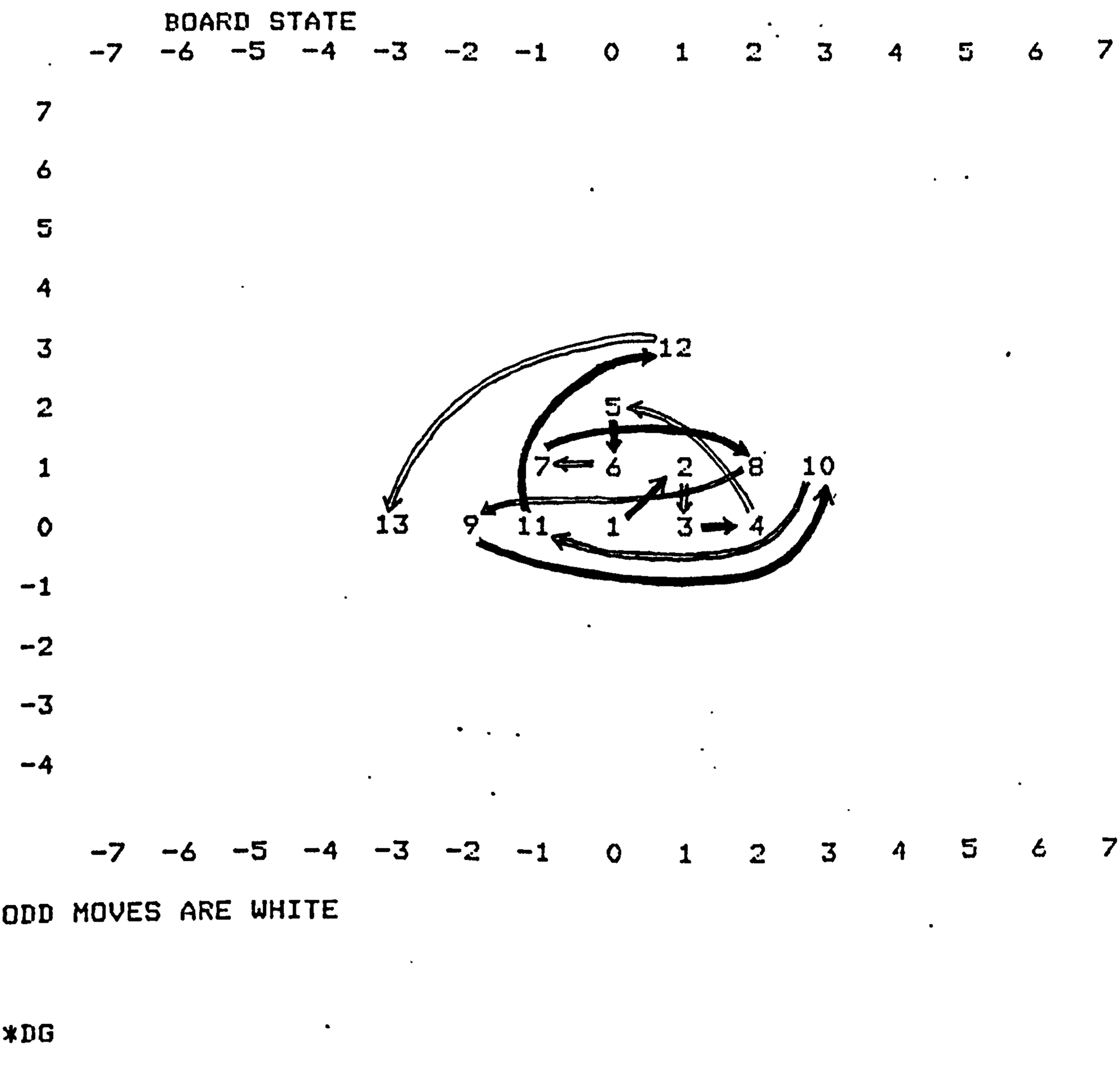


Figure 4.13(a) Move Sequence (Both Defending)

```
*
*
*DS
```



ODD MOVES ARE WHITE

```
*DG
```

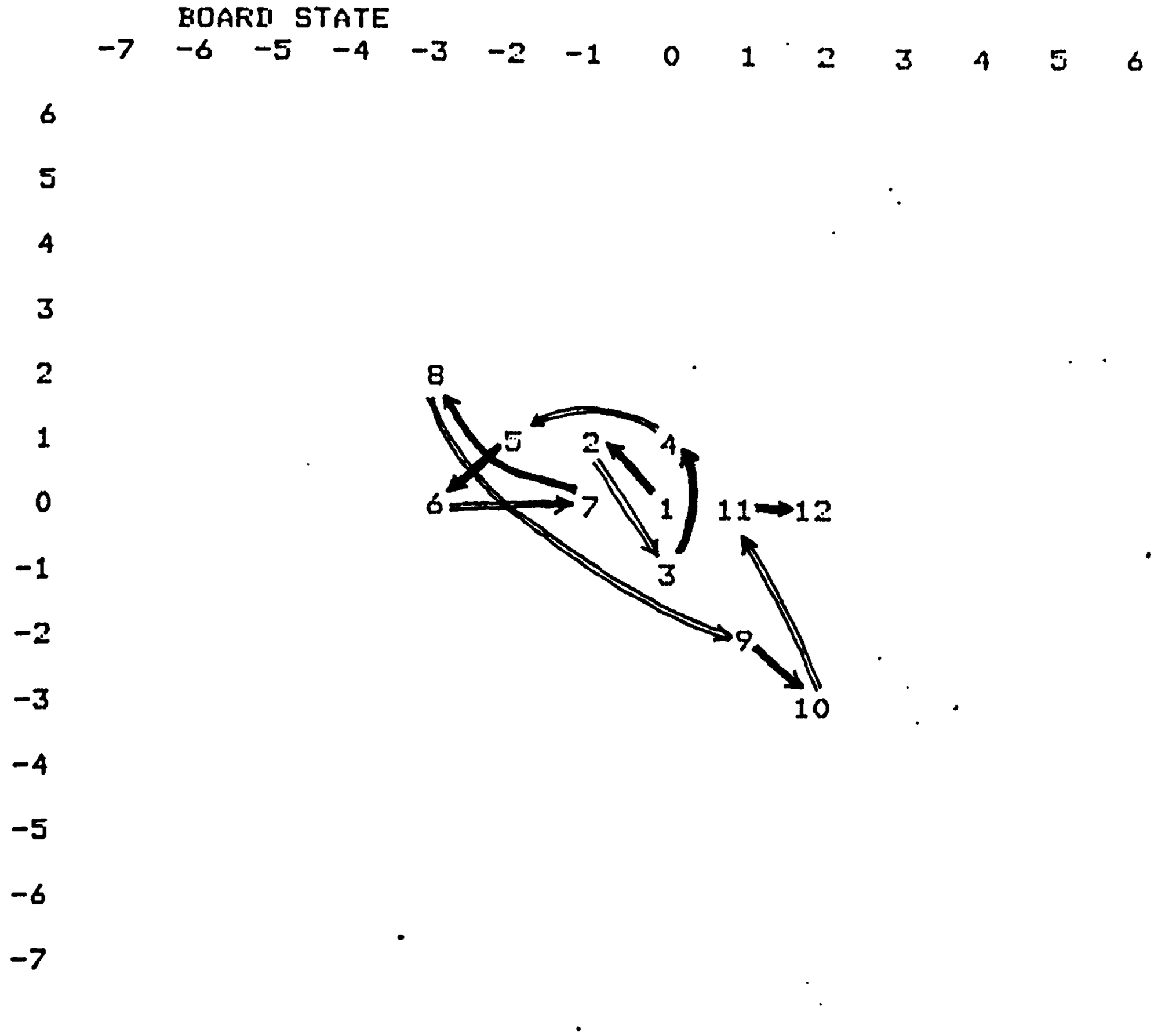
```
          BOARD STATE
    -7  -5  -3  -1   1   3   5   7

 7          .       .       .
 6        .   .   . .     .
 5      . . . . . . . . . . . . .
 4    . . . . . . . . . . . . . .
 3      . . . . . . . B . . . .
 2        . . . . . W . . . .
 1        . . . . W B B B B . . . .
 0    . . . . W W W W W B . . . .
-1          . . . . . . . . . .
-2        . . . . . . . . . . . .
-3      . . . . . . . . . . . . . .
-4    . . . . . . . . . . . . . .

    -7  -5  -3  -1   1   3   5   7
```

Figure 4.13(b) Move Sequence (Both Attacking)

*DS



ODD MOVES ARE WHITE

*DG

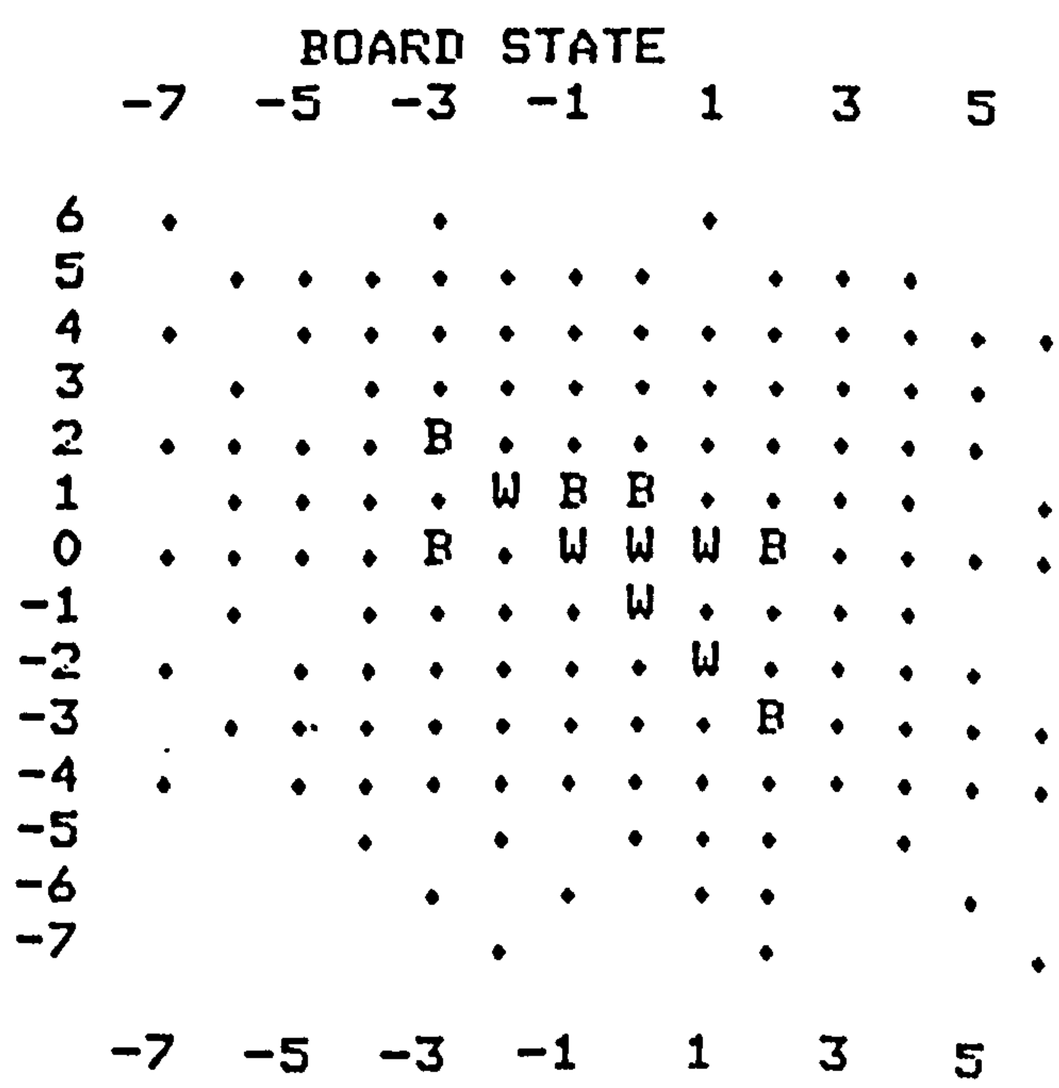
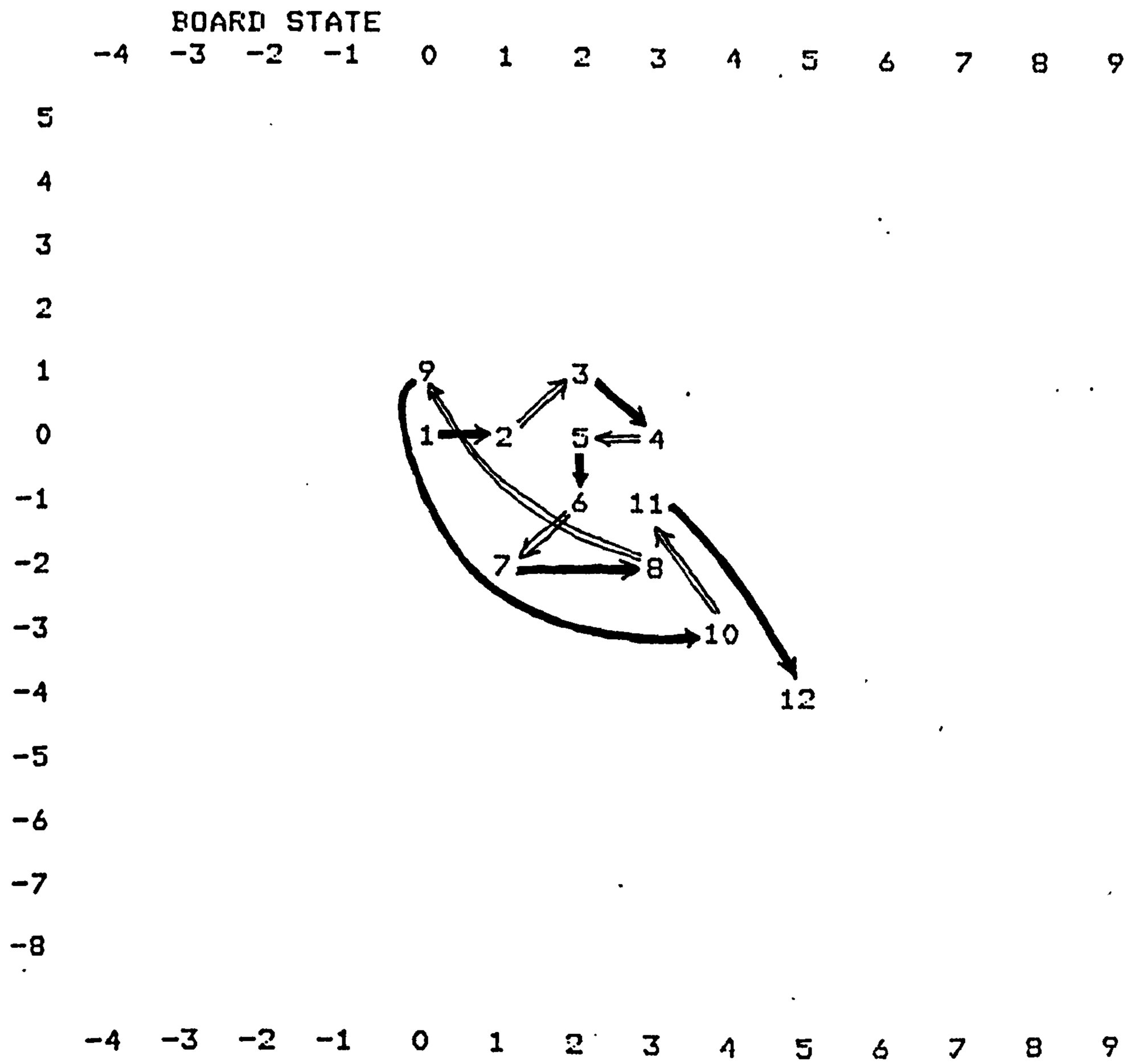


Figure 4.13(c) Move Sequence (White Attacking, Black Defending)

*DS



ODD MOVES ARE WHITE

*DG

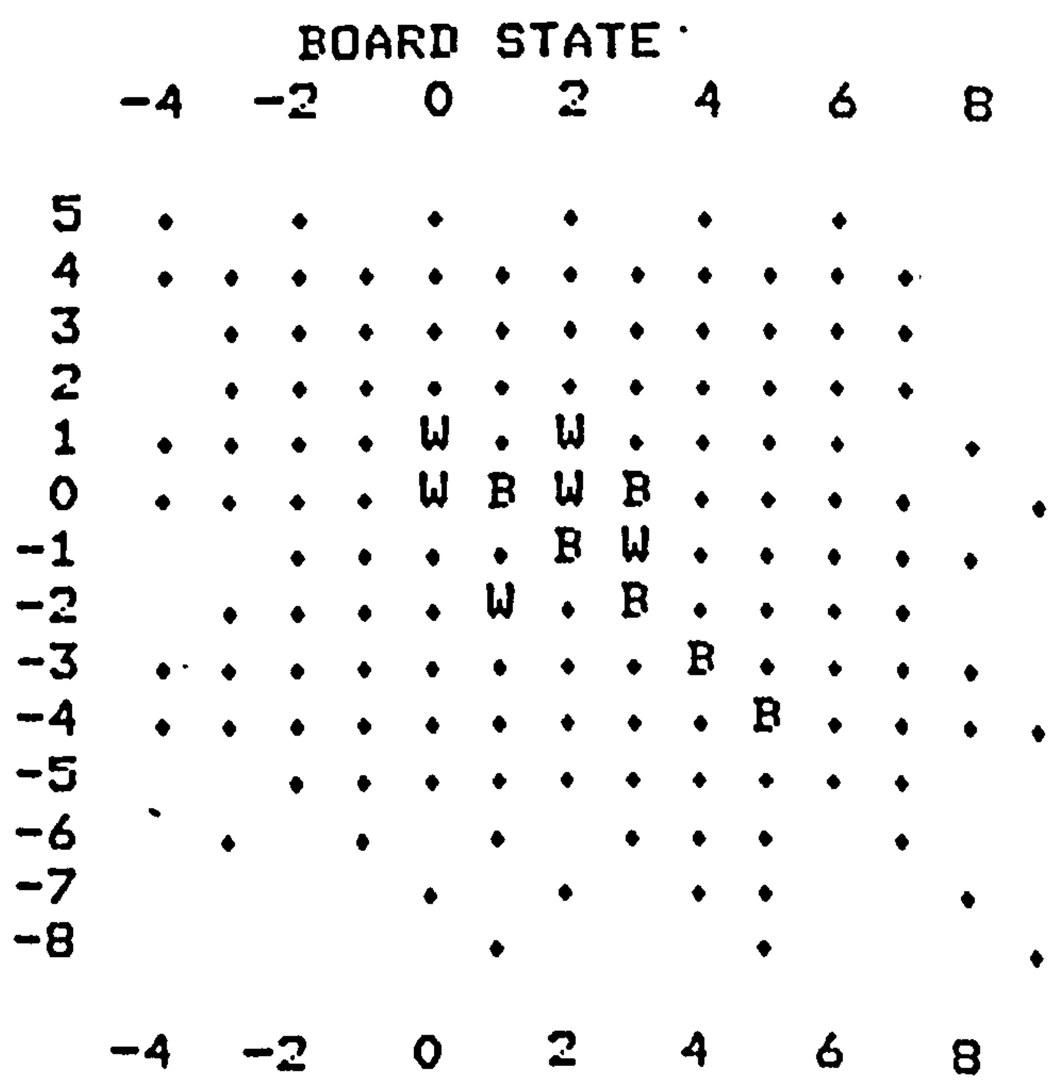
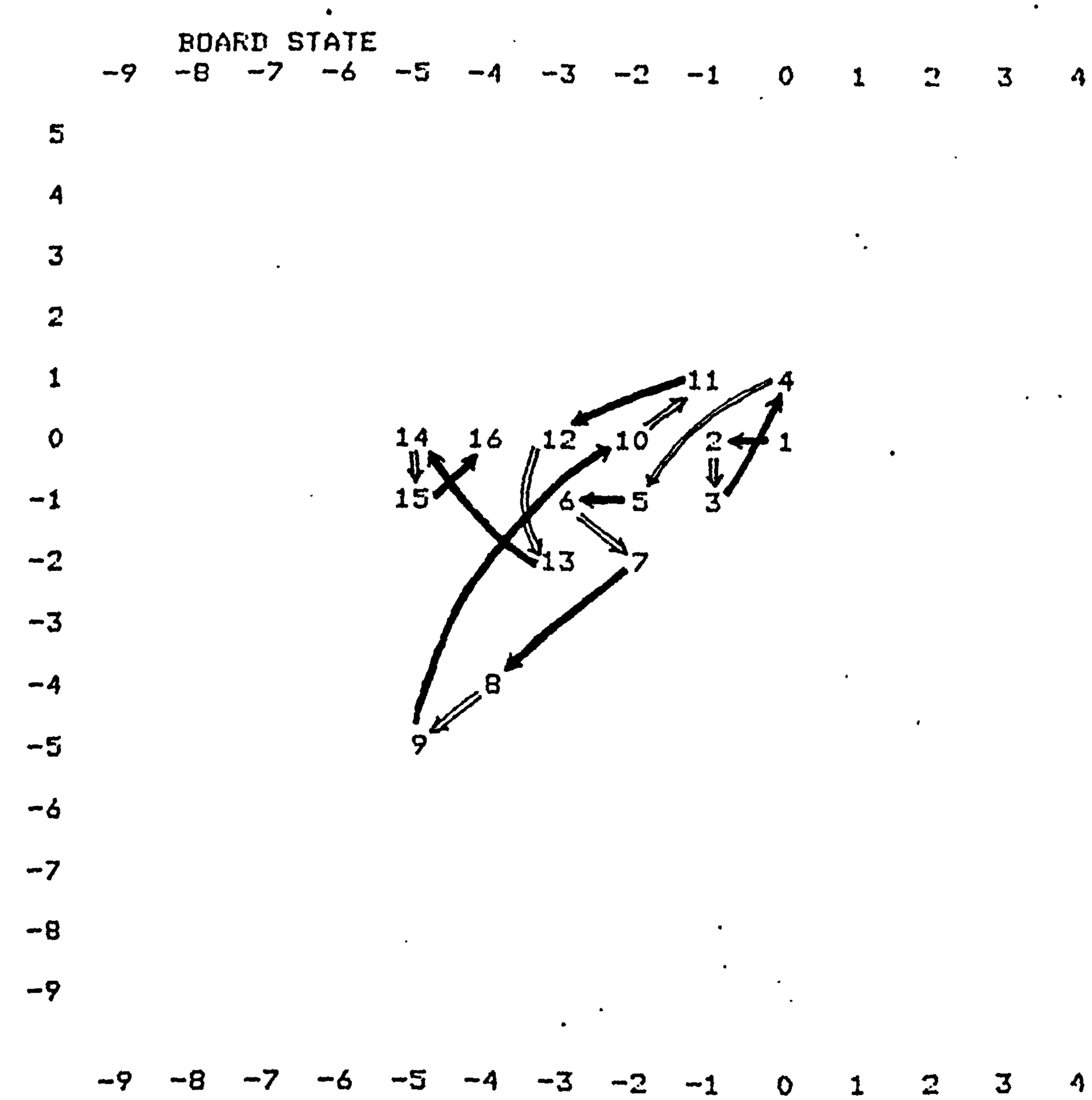


Figure 4.13(d) Move Sequence (White Defending, Black Attacking)

*DS



ODD MOVES ARE WHITE

*DG

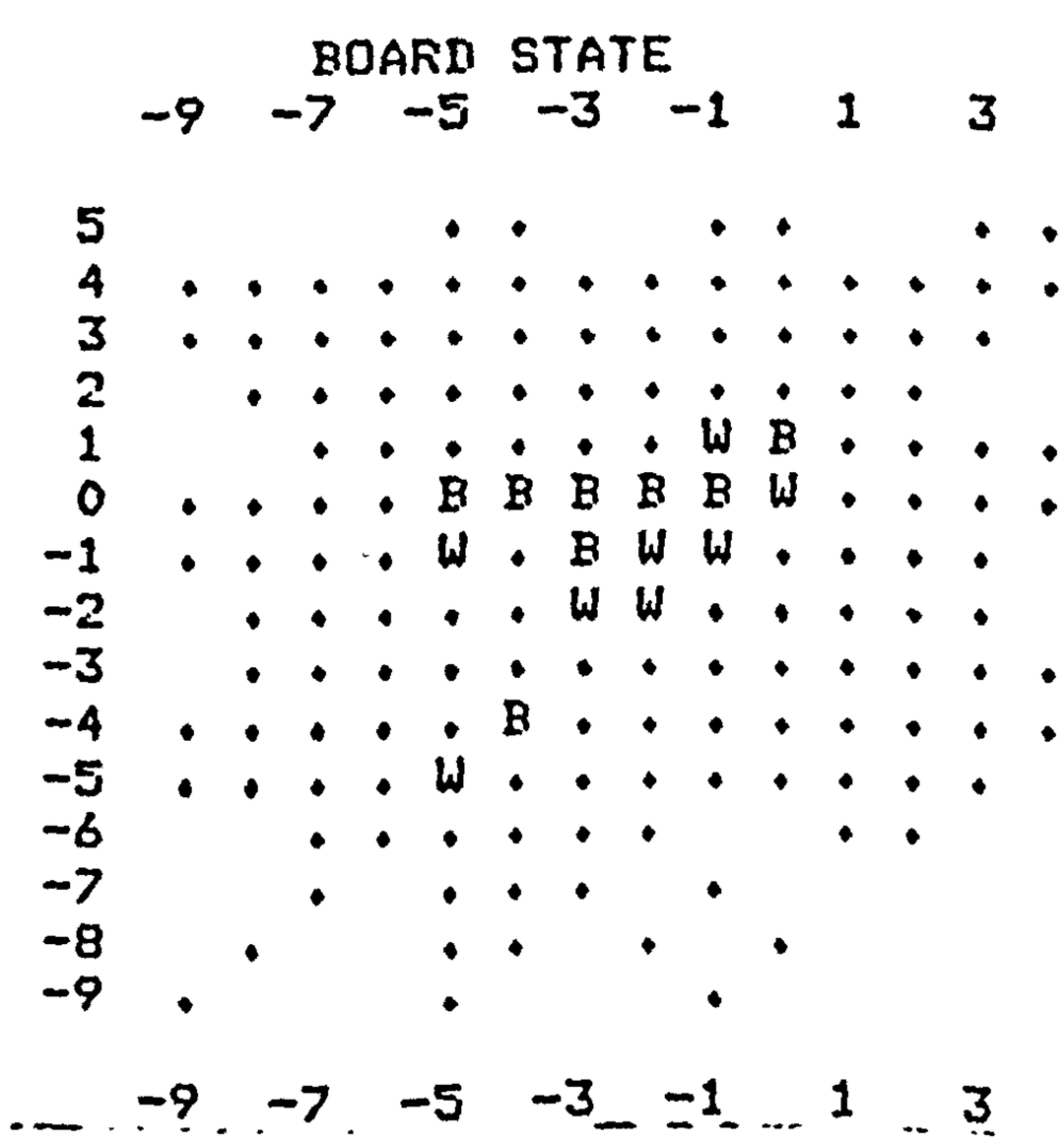


Figure 4.13(e) Move Sequence (Dynamic Defending/Attacking)

White moves first so all odd moves (odd numbers) are White and even moves Black. The sequence of moves is emphasised by the arrows. Fig. 4.13 has the following sections:

a) MAX vs. MAX, both DEFENDING

b) MAX vs. MAX, both ATTACKING

c) MAX vs. MAX, White ATTACKING, Black DEFENDING

d) MAX vs. MAX, White DEFENDING, Black ATTACKING

e) WA.140 vs. WA.140, dynamic DEFENDING and ATTACKING

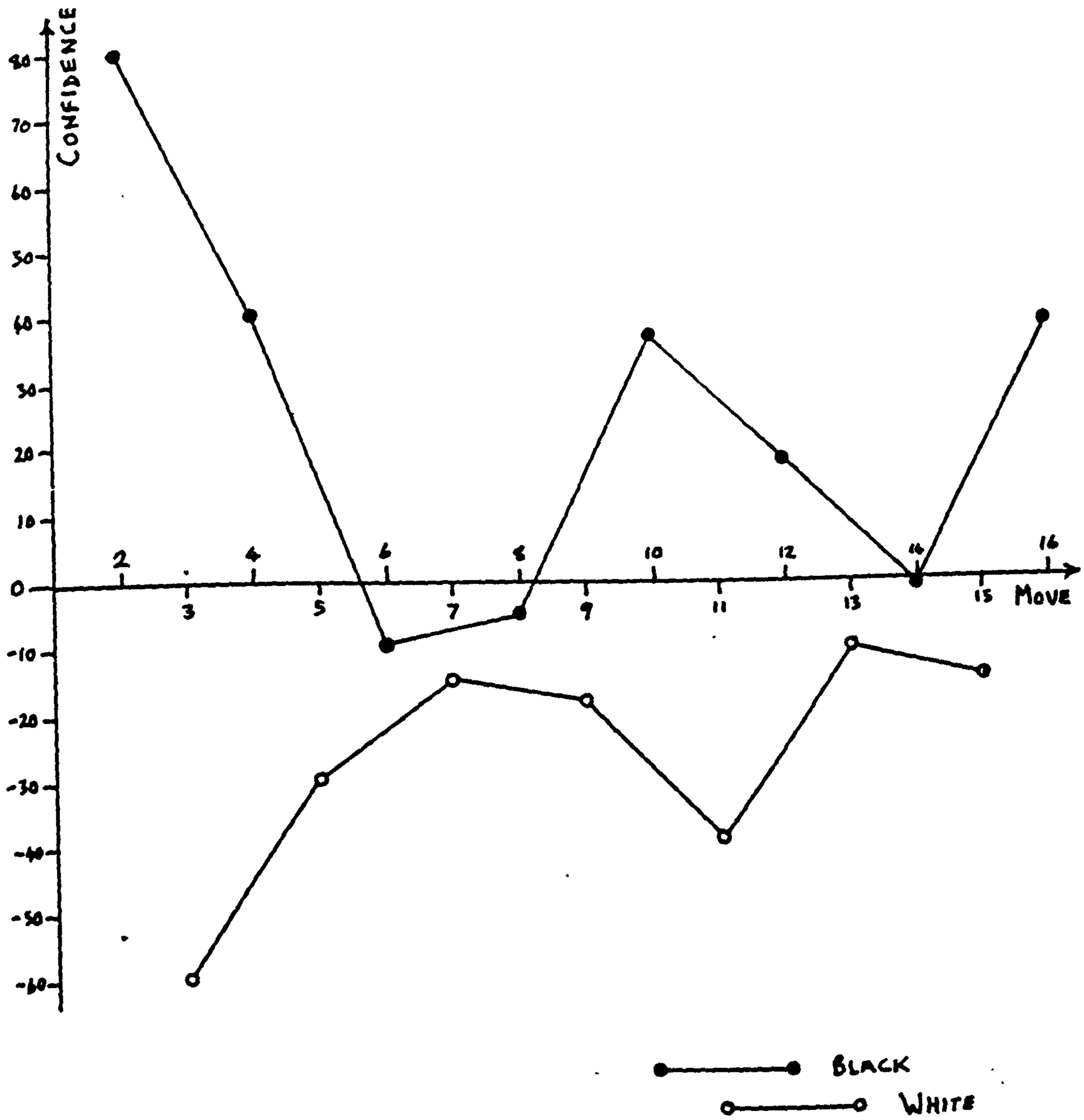
"MAX" refers to the fully trained model with the line ranking given in fig. 4.14. "WA.140" refers to the state of White after 140 training moves against an untrained, non-learning, opponent. MAX is thus a fully trained player and WA.140 is an arbitrary partially trained player. (The training of WA.140 can be found in the tables of figs. 4.8(1), 4.9(1), 4.10(1) and 4.11(1).)

With both players defending (fig. 4.13(a)), each player follows his opponent, blocking their moves, but neither succeeds in forming their own winning line. Fig. 4.13(b) shows that if both players attack, each player plays his own game with little regard for his opponent. The first to start wins. With one player attacking and the other defending (figs. 4.13(c) and 4.13(d)) the attacker concentrates on forming his winning line while the defender concentrates on blocking him. Both attacking and defensive play are exhibited in fig. 4.13(e) which is the mode of play with defensive and attacking play controlled dynamically as explained in section 4.8.3 below.

Pattern	Address	Line Value
XXXXX	31	19
XXXX-	30	18
-XXXX	15	18
XXX-X	29	17
X-XXX	23	17
XX-XX	27	16
-XXX-	14	15
XXX--	28	14
--XXX	7	14
X-XX-	22	13
-XX-X	13	13
XX-X-	26	12
-X-XX	11	12
XX--X	25	11
X--XX	19	11
X-X-X	21	10
-XX--	12	9
--XX-	6	9
XX---	24	8
---XX	3	8
-X-X-	10	7
X-X--	20	6
--X-X	5	6
X--X-	18	5
-X--X	9	5
X---X	17	4
--X--	4	3
-X---	8	2
---X-	2	2
X----	16	1
----X	1	1
-----	0	0
	Total	306

Address	Value
0	0
1	1
2	2
3	8
4	3
5	6
6	9
7	14
8	2
9	5
10	7
11	12
12	9
13	13
14	15
15	18
16	1
17	4
18	5
19	11
20	6
21	10
22	13
23	17
24	8
25	11
26	12
27	16
28	14
29	17
30	18
31	19

Figure 4.14 Fully Trained Line Evaluation Element - MAX



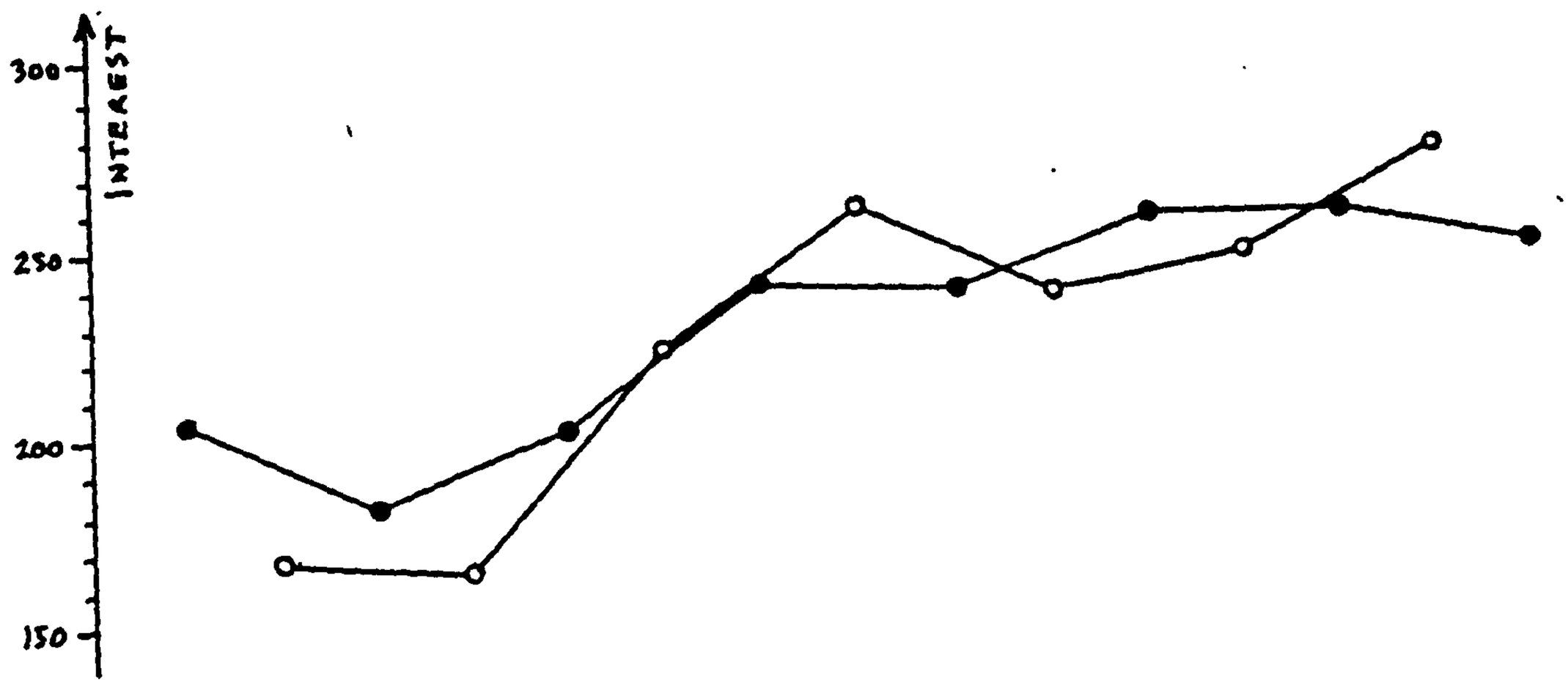


Figure 4.15 Confidence & Interest Graph (During Fig. 4.13(e))

4.8.2 Confidence Evaluation

In order to allows dynamic control of the DEFENDING and ATTACKING parameters a CONFIDENCE parameter is maintained giving a basis for setting DEFENDING and ATTACKING (section 4.8.3).

The player evaluates his opponent's and his own previous move using the move evaluation of section 3.4.3.2. A new confidence level is then calculated before each move from the player's previous confidence and the values of these previous two moves:

```
new CONFIDENCE = 1/2 (previous CONFIDENCE
                      + value of player's previous move
                      - value of opponent's previous move)
```

A graph of the levels of CONFIDENCE of two players during a game is given in fig. 4.15. This game is depicted in fig. 4.13(e), see section 4.8.1 above.

4.8.3 Setting of DEFENDING and ATTACKING

The DEFENDING and ATTACKING parameters are set before each move according to the following two conditional expressions:

```
DEFENDING = true if DEFENDING and CONFIDENCE < DOFF
                 or if not DEFENDING and CONFIDENCE < DON
          = false otherwise
ATTACKING = true if ATTACKING and CONFIDENCE > AOFF
                 or if not ATTACKING and CONFIDENCE > AON
          = false otherwise
```

DOFF, DON, AOFF and AON are hysteresis switching values, set by the teacher. Normally AOFF <= DON <= AON <= DOFF for hysteresis and to ensure either ATTACKING or DEFENDING or both are true for all changes in CONFIDENCE.

All results presented in this thesis use

AOFF = DON = AON = DOFF = 0

giving attacking play in the positive (upper) half of fig. 4.15 and defending play in the negative (lower) half. Moves taken while CONFIDENCE is at zero have both DEFENDING and ATTACKING parameters on. See sections 3.4.4.1 and 3.4.4.6 and Boxes 1 and 6 of the move search flow chart (fig. 3.12) for the use of DEFENDING and ATTACKING parameters.

No investigation of the effects of varying these hysteresis thresholds was done.

4.8.4 MEDIOCRE and OUTSTANDING Thresholds

The heuristic of section 3.4.3.3 used to evaluate a move in terms of the values of the lines to which it contributes has two thresholds which determine the way a value is derived for the move.

The OUTSTANDING parameter determines when a single line, containing the move is, considered sufficiently "outstanding" to characterise, by itself, the value of the move. The term "outstanding", and hence the parameter, can imply either that the line is valued higher than a given (absolute) threshold value, or that it is valued higher than any other line containing the move by an amount greater than a given (relative)

threshold. The existing implementation takes the former interpretation and is programmed as follows:

During move evaluation (section 3.4.3.2) the value of a move is taken to be 20 times the value of best line intersecting at the move if the best line has a value greater or equal to the OUTSTANDING threshold. If the best line has a value less than the OUTSTANDING threshold then the move value will be formed from the sum of values of intersecting lines which are greater or equal to the MEDIOCRE threshold.

The multiplication of the best line value by 20, in the case of an "outstanding" line, ensures that the corresponding move value will be higher than any move value derived by summation of up to 20 "mediocre" line values.

The MEDIOCRE threshold causes very low line values (those below the MEDIOCRE threshold) to be excluded from move evaluations, thus avoiding moves surrounded by many such low valued lines being valued higher than moves surrounded by a few higher valued lines. This essentially reflects a characteristic of Kenju which is that the "strength" of a move usually depends on one or two "good" intersecting lines rather than the combination of many intersecting lines.

Due to the nature of the training of the line evaluation net (section 4.6) whereby line values are continually increasing with further training, no satisfactory basis has been found for the setting of the MEDIOCRE and OUTSTANDING parameters.

If these thresholds are given fixed values (see later, section 4.8.4.1, figs. 4.16 and 4.17) they will only have the desired effect at one point

during training. For example, if they are set according to the values of a completely trained line evaluation net, then they will only have their desired effect after the line evaluation net has been fully trained. The other than desired effect of these thresholds during this training may be detrimental to the computer's play, and possibly to the training of the line evaluation net itself.

If the MEDIOCRE and OUTSTANDING thresholds are set according to certain fixed proportions of the current values of lines (given by the line evaluation) then the above problem is alleviated. See later, Fig. 4.18, section 4.8.4.1.

For example, the thresholds could be set on the basis of two percentages of the highest value in the line evaluation net (i.e. the value of the top ranking line). These two percentage levels (one for MEDIOCRE, one for OUTSTANDING) could then be chosen so as to give the desired MEDIOCRE and OUTSTANDING threshold effect when applied to the fully trained net (fig. 4.14). This latter scheme might then have the desired, or near desired, effect with a partially trained net. That is, lines valued at x% of the max line value and over would be considered at least MEDIOCRE while those valued at y% or more of the max line value would be considered OUTSTANDING by the move evaluation heuristic (section 3.4.3.3).

The experimental investigations presented in section 4.9 do not use these proportional settings of MEDIOCRE and OUTSTANDING but the extreme settings discussed now.

Extreme settings can determine the degree of effect which might be expected for less than optimal settings of the MEDIOCRE and OUTSTANDING

thresholds. This has particular relevance if these thresholds are set dynamically by the learning algorithm itself. The present system allows only for the teacher to set the two thresholds.

The possible extreme settings are:

MEDIOCRE	OUSTANDING	effect
0	0	A
0	∞	B
∞	0	C
∞	∞	D

A and C have the same effect of always taking 20 times the best intersecting line value as the value of the move. B has the effect of always taking the sum of all intersecting line values as the value of the move. D has the trivial effect of always valuing moves at zero. The ∞ setting can be taken as any large value greater than the highest line value possible (from line evaluation).

```
*DVW

          BOARD STATE
     -6  -5  -4  -3  -2  -1   0   1   2   3   4   5   6   7

  6  80                 80  80                 80  80                 80
  5  80 100               80 220               220  80             100  80
  4      100 140           260 140 140 260           200 220
  3  80  80 120 180 340 220 180 300 260 340                         80
  2  80 100 140 180   B 260 180   W 380 140 100  80 220
  1  80 100 140 180   B 300 280   W 180 140 100 260
  0      80 220 260 360   W   W 300 260 220 300
 -1  80 100 140 180   B   W 160 260 180   B 180 140 100  80
 -2  80 100 140 180   W 300   B 300   B 260 200 100  80
 -3  80     120 380 160 260 180 300 180 180     140
 -4     100 340     140 220 260 100 140 100 140     100
 -5  80 220     100 100 220 100 100 100 100     100     80
 -6  80     80     80     80     80     80     80

     -6  -5  -4  -3  -2  -1   0   1   2   3   4   5   6   7


*DG

          BOARD STATE
     -6  -4  -2   0   2   4   6

  6  .     . .     . .     .
  5  . .   . .     . .   . .
  4    . .   . . . .   . .
  3  . . . . . . . o . .       .
  2  . . . . B . . W . . . . .
  1  . . . . B . . W . . . .
  0    . . . . W W b . . .
 -1  . . . . B W . . . B . . . .
 -2  . . . . W . B . B . . . .
 -3  .   . c . . . . . .   .
 -4    . .   . . . . . . .   .
 -5  . .   . . . . . . .   .   .
 -6  .   .   .   .   .   .   .

     -6  -4  -2   0   2   4   6


*
```

Figure 4.16 Effect A/C

*DVW

BOARD STATE

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
6	4			4	4			4	4			4		
5	4	10		4	26			26	4		10	4		
4		5	13		41	23	23	41		23	16			
3	4	4	6	17	54	63	63	51	30	39				4
2	8	20	24	34	B	88	53	W	89	29	14	4	16	
1	4	20	38	66	B	82	62	W	70	48	10	35		
0		4	54	99	91	W	W	102	85	69	55			
-1	4	15	25	42	B	W	29	86	60	B	44	19	10	4
-2	8	20	13	25	W	105	B	59	B	76	40	14	4	
-3	4		6	59	8	99	36	63	39	36		13		
-4		5	35		25	16	63	10	25	14	13		10	
-5	4	16		10	5	20	10	10	14	9		10		4
-6	4		4		8		4		8		4		4	
	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7

*DG

BOARD STATE

	-6	-4	-2	0	2	4	6
6	.	. .	. .	. .	.	.	
5	. .	. .	. .	. .	.	. .	
4	. .	. .	. . . .	.	. .		
3	. . . .	. . .	. . .	a . .	.		.
2	. . . .	B . .	. . .	W . .	. . .	. .	
1	. . . .	B . .	. . .	W . .	. . .	.	
0	. . . .	. W	W . .	b . .	.		
-1	. . . .	B W	. . .	. B .	. . .	.	.
-2	. . . .	W .	B .	B . .	. . .	.	
-3	. .	c .	. . .	. . .	.	.	
-4	. .	. .	. . .	. . .	.	.	
-5	. .	. .	. . .	. . .	.	.	.
-6	. .	. .	. .	. .	.	.	
	-6	-4	-2	0	2	4	6

*

Figure 4.17 Effect B

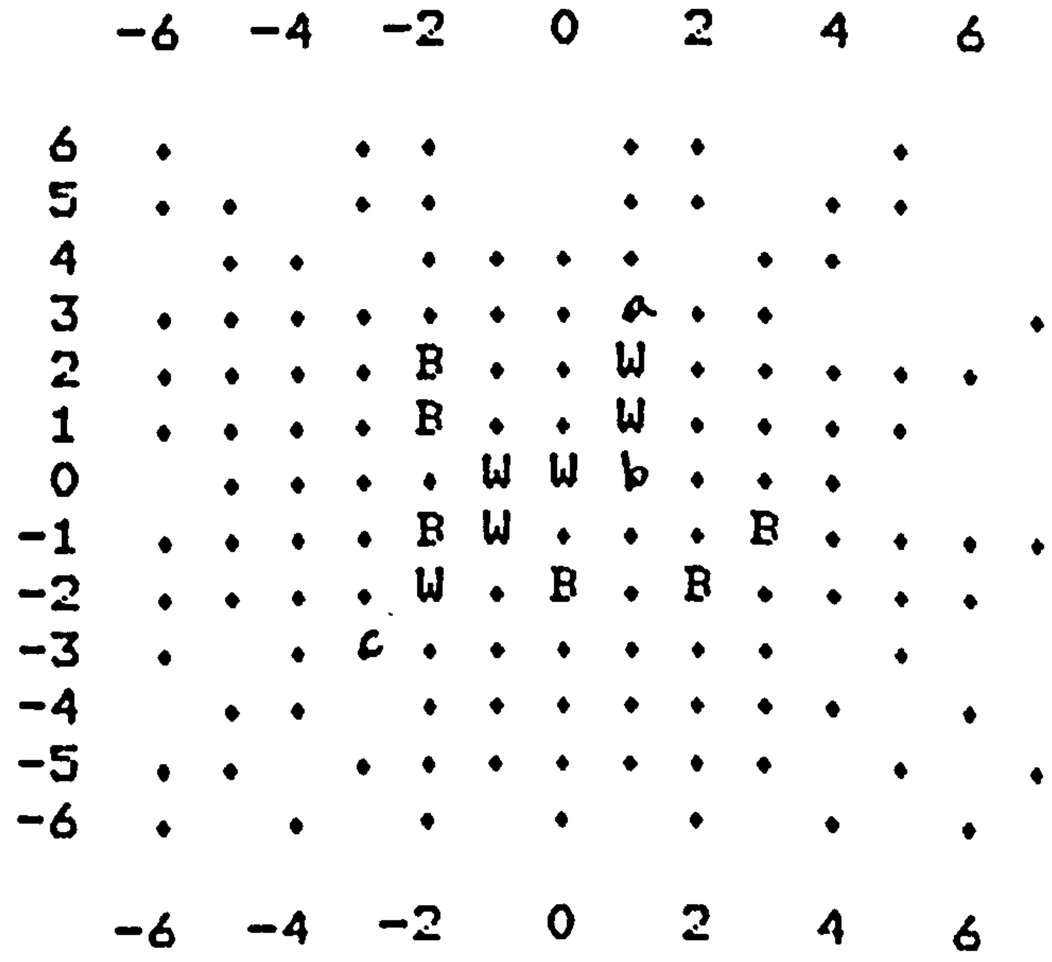
*DVW

BOARD STATE

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
6														
5					11			11						
4					25			25		10	11			
3				9	46	29	18	43	22	29				
2				18	B	52	18	W	380				11	
1				36	B	52	32	W	36			25		
0			11	61	75	W	W	86	25	29	43			
-1				18	B	W		43	36	B	18			
-2				9	W	73	B	40	B	52	19			
-3				380		61	9	43	18	9				
-4			29			11	25							
-5		11				11								
-6														

*DG

BOARD STATE



*

Figure 4.18 Percentage Effect

4.8.4.1 Results of MEDIOCRE and OUTSTANDING Settings

Figs. 4.16 and 4.17 illustrate the effects of these extreme settings with the fully trained net of fig. 4.14. They thus represent the best that the move evaluation algorithm can be expected to achieve with these extreme settings.

The same moves are evaluated with the MEDIOCRE threshold set at 47% and OUTSTANDING at 100% of the maximum line value as shown in fig. 4.18.

The effect of A and C can be seen in fig 4.16. Note that the formation of two intersecting threes by move b in fig. 4.16 is valued no more than the formation of a single three by move a. Thus the playing algorithm will be "blind" to the difference between moves of more than one intersecting line and moves of a single intersecting line (see fig. 2.2 and Appendix A).

Move c in fig. 4.16 is however correctly valued as the best move (it completes a winning line).

The effect of B is illustrated in fig. 4.17. Move b is now correctly evaluated as better than move a since move b is the sum of both horizontal and vertical line values, but move c is no longer valued as the best move since it is not at the intersection of many lines.

The percentage scheme of fig 4.18 correctly values c as the best move with b valued higher than a. The percentages were specifically chosen to achieve correct evaluation with the fully trained line evaluation net of fig. 4.14.

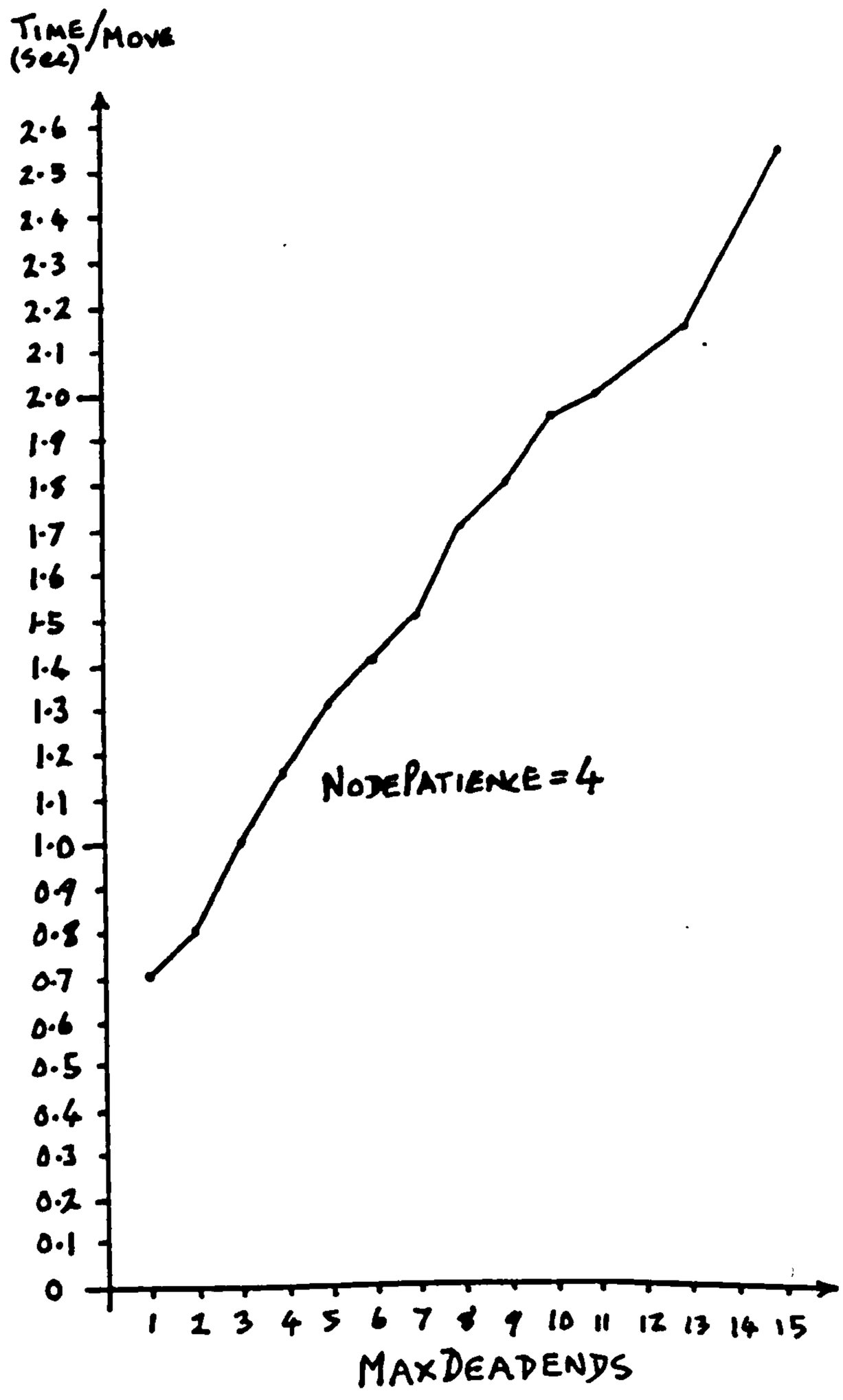


Figure 4.19 MAXDEADENDS Timings

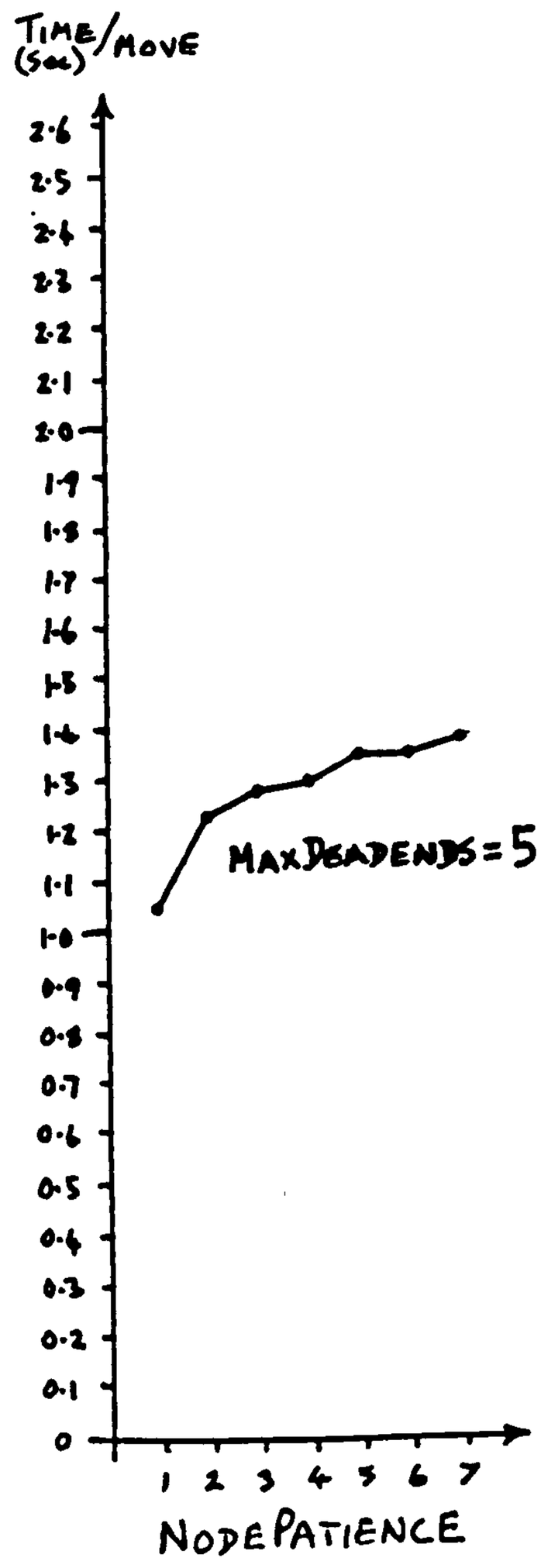


Figure 4.20 NODEPATIENCE Timings

4.8.5 The NODEPATIENCE and MAXDEADEND Parameters

The parameter which controls the number of iterations of the search loop (section 3.4.4.4 and fig. 3.12) and hence the number of nodes in a search path (section 3.4.5 and figs. 3.13 and 3.14) is the NODEPATIENCE parameter. The search path may be foreshortened by reaching a dead-end or looping condition (section 3.4.4.5). If the search path terminates due to the iteration patience parameter (NODEPATIENCE) the search as a whole terminates "successfully" with the best move of that path. If a dead-end (i.e. looping) is encountered during a search path it is abandoned and a new search path started. The maximum number of dead end search paths followed is controlled by the MAXDEADENDS parameter (section 3.4.4.7). Thus NODEPATIENCE controls the length of successful paths which do not end with dead-end loops, and MAXDEADENDS limits the number of unsuccessful, dead-end, paths followed.

These two "patience" parameters are set by the teacher in the existing system but, as pointed out in section 3.4.4.7, the simulated player's "interest" at any particular time could determine their setting. They might also be set according to a time constraint.

The graphs of figs. 4.19 and 4.20 show the average time taken for a move with varying settings of MAXDEADENDS and NODEPATIENCE. MAXDEADENDS has the major effect on move search time because most search paths are dead-end paths.

If these parameters were to be set dynamically by the algorithm itself MAXDEADENDS could give a coarse control of the extent of the search and

the time taken, while NODEPATIENCE could provide a fine control.

The "interest" of the player at a particular stage of a game refered to above, might be based on a running sum of the values of the opponent's and player's previous moves (the previous two moves) calculated before each move search in a similar way to the CONFIDENCE parameter of section 4.8.2:

```
new INTEREST = 1/2 (current INTEREST
                    + value of player's previous move
                    + value of opponent's previous move)
```

A graph of the INTEREST level during a game is given with CONFIDENCE levels in fig. 4.15.

A problem arises in using the INTEREST level directly as a control for the "patience" parameters in that it depends on the degree of training of the line evaluation net (sections 3.4.3.1, 4.6 and 4.7). This is similar to the problem encountered with setting the thresholds in the move evaluation heuristic (sections 3.4.3.3 and 4.8.4) and might again be partially solved by normalising the parameter to take account of the level to which the line evaluation net has been trained.

The INTEREST parameter plays no part in the setting of other parameters in the existing implementation.

4.8.6 Computer Player vs. Computer Opponent Capability

In order to implement the mode of learning where a computer player plays against a computer opponent, as suggested in the section on modes of

learning (section 4.3), the system must have the ability to swap between simulating one player and the other.

To allow this, two copies of the player's parameters and RAM net (line evaluation element) are maintained, one for each player. Between each move the active parameters and RAM net are changed to be those of the player who is to take the next move.

This ability is particularly useful in that it allows computer players with differing parameter settings and experience (RAM net contents) to be assessed against a standard opponent (discussed more fully in the section on training schemes, section 4.9).

4.9 Training Schemes

The training schemes given in fig. 4.21 produce the changes in White's line evaluation net which are tabulated in figs. 4.8, 4.9 and 4.10 and shown graphically in fig. 4.11. The first four training schemes have White learining against a non-learning opponent while scheme 5 has White and Black both learning and so each is learning while playing an opponent who is himself improving. Scheme 6 represents an example of learning mode (a) from section 4.3, where both White and Black learn while observing the play of two human players (both players are the author in this case). Neither White nor Black take moves or effect the course of the game in any way with mode (a).

Scheme 4 corresponds to mode (b) of section 4.3 and schemes 1, 2, 3, and 5 to mode (c).

Schemes 1 and 2 train White against a totally untrained opponent (NULL) and a fully trained opponent (MAX) respectively. The untrained opponent has its line evaluation net set to zero for all line patterns and the fully trained opponent has the line evaluation net contents of fig. 4.14. Scheme 3 trains White against a partially trained opponent (WA.140) selected arbitrarily from the results of another training scheme (scheme 1).

This set of training schemes was chosen to show the influence of different opponents (with varying degrees of training) on the player's learning. The schemes with the author as one or both of the players (schemes 4 and 6) were included to investigate the influence of a human player (opponent) on learning.

1)	WA	W learns from scratch vs. NULL
2)	W6	W learns from scratch vs. MAX
3)	W8	W learns from scratch vs. WA.140
4)	WC	W learns from scratch vs. ME
5)	W4	W+B learning from scratch W vs. B
6)	WE	W+B learning from scratch ME vs. ME

Fig. 4.21 Six Training Schemes

a) Parameters are set as in fig. 4.23.

b) The initial contents of the line evaluation net are zero.

c) TRYING and LEARNING are both turned on. CONFIDENCE is reset to zero before each game.

d) Teaching advice is according to the fully trained line evaluation net of fig. 4.14 (i.e. MAX).

e) Training takes place during as many games as is required to give each player experience of taking 200 moves (i.e. 200 moves each).

f) White always takes the first move of each game.

Fig. 4.22 Training Method for Section 4.9 Results

MAXDEADENDS = 5

NODEPATIENCE = 4

MEDIOCRE = 0

OUTSTANDING = 0

TRYING = True

LEARNING = True

DON = 0

DOFF = 0

AON = 0

AOFF = 0

Fig. 4.23 Parameter Settings for Section 4.9 Results

Except for the training corresponding to the learning curves discussed in section 4.7.1 for which figs. 4.6 and 4.7 apply, all training is done according to fig. 4.22 and 4.23.

Note that MEDIOCRE and OUTSTANDING are both set to zero giving effect A (section 4.8.4) which reduces the move evaluation heuristic (section 3.4.3.3) to one which values a move as 20 times the value of the highest valued line intersecting at the move. This reduced heuristic is a function of only one intersecting line and hence gives no weight to intersections of more than one line pattern. Effect A is chosen in preference to effect B (the other usable effect of extreme settings of MEDIOCRE and OUTSTANDING, section 4.8.4) which would reduce the move evaluation heuristic to one which values a move as the sum of the values of all lines intersecting at the move position. The correct valuing of

the winning move, c, in fig. 4.16 is considered more important than the correct valuing of the intersection of two open threes at move b (at the cost of incorrect valuing at move c) in fig. 4.17. Refer to section 4.8.4 for a full explanation of the effect of extreme settings of MEDIOCRE and OUTSTANDING.

The patience parameters (MAXDEADENDS and NODEPATIENCE) are set almost arbitrarily. The values are chosen, on an intuitive basis, to be large enough to give a reasonable variety of search path characteristics. Fig. 4.24 presents these characteristics in terms of search path lengths and types of termination of paths.

The effect of the patience parameters could be given a much more extensive analysis than seems to be required here.

The hysteresis effects of the DEFENDING and ATTACKING switching parameters, DOFF, DON, AON and AOFF, have not been investigated (see section 4.8.3). Setting them all at zero (fig. 4.23) gives defensive play when CONFIDENCE is negative, attacking play when CONFIDENCE is positive, and the possibility of both defensive and attacking play when CONFIDENCE is zero. Widening the hysteresis could allow a greater overlap of defensive and attacking play giving less radically defensive or attacking play.

The game depicted in fig. 4.13(e) illustrates the dynamic nature of the DEFENDING and ATTACKING parameters with the hysteresis parameters set to zero. Fig. 4.15 gives a graph of CONFIDENCE during the same game.

Paths Tried per Search		Node Patience 1	4	9
	5	78.1%	82.7%	90.1%
	4	6.6%	7.1%	4.8%
(Max Deadends = 5)	3	5.0%	4.6%	2.2%
	2	4.0%	4.6%	1.4%
	1	6.3%	1.0%	1.4%

Deadend Cause	Node Patience 1	4	9
Node Loop deadend	67.0%	74.9%	76.8%
Line Loop deadend	19.3%	14.4%	16.2%
Place Loop deadend	1.0%	2.8%	1.5%
No deadend	12.7%	7.9%	5.4%

Figure 4.24 Deadends Statistics

4.10 Learning in System 4

4.10.1 Introduction

Learning in system 4 follows the same lines as learning in system 3 (section 4.5) except that the teacher is asked for advice about moves rather than lines (as in system 3). See section 3.5 for a discussion of the difference between the two systems and the philosophy behind it.

4.10.2 Training the Line Evaluation Net

The search algorithm of system 4 asks the teacher for advice as to which is the better of two moves valued equal whilst finding the best of the set of moves in a line during a move search (see section 3.4.4.2). That is, during the process of finding the best move in a line (Box 2 in the move search flowchart of fig. 3.12), each move in turn is compared by value with the current best move in the line. If any move is found to be equal in value to the current best move then the system asks the teacher which of the two conflicting moves should be considered the better move and hence the current best in the line. In answer to this query the teacher indicates the better move or, when appropriate, that the moves are in fact correctly valued equal.

All line patterns found intersecting with the preferred move, but not found among the lines intersecting with the other move, are "rewarded" by incrementation in the same way as in system 3 (section 4.6). That is, the value given by the simple line evaluation net of section 3.4.3.1 to each of these preferred line patterns is incremented, thus ranking these lines one higher than previously. Hence, the net will subsequently value

lines with these line patterns one higher than before.

The learning implicit in each piece of advice about moves changes the stored knowledge relating to several of the lines which intersect at the preferred move. Therefore, the rate at which the system learns might be expected to be greater than that for line oriented learning in system 3.

However, the assumptions made in sections 4.6 and 4.7.6 about line ranking asymptotically approaching that of the fully trained net of fig. 4.14 would be violated. This is because lines could be re-ranked erroneously, which was not possible in system 3, given correct and consistant advice from the teacher.

4.11 Some Comments on this Chapter

Chapter 4 does three things:

1) The learning aspects of the four systems are outlined, pointing out the development from the initial system through to system 4.

2) It searches for various means of measuring system behaviour. An external viewpoint is taken with the performance graphs (section 4.7.1) with little success. Attention is switched to a study of the internal mechanisms with special regard to the learning element.

3) Through these measurements it assesses the effect of some parameters and training schemes.

Overall, the variability of play, which is a consequence of the sequential nature of the game, leads to widely varying results making analysis difficult.

Nevertheless, the results from the experiments with the ATTACKING and DEFENDING parameters are as expected. This indicates a correctly functioning algorithm.

The planned comparison of different training methods was thwarted by the lack of outstanding differences attributable to differences in the training schemes.

CHAPTER 5

DISCUSSION AND CONCLUSIONS

5.1 Summary of Chapters

5.1.1 Chapter 1

Chapter 1 first states the aim of this thesis as an investigation of a learning process modelled as a computer program. Although a computerised game-player is chosen as the vehicle for research it is made clear that no contribution to computer game playing is claimed (or was attempted). However, since much of the work described in the thesis is concerned with developing a game-playing program, a brief background to the art of computer game-playing is given. Emphasis is placed on a few examples of work throwing light on how humans play games, at the expense of an exhaustive study of computer game-playing techniques. In this the recognition of patterns in a game and the ability to learn are prominent.

A short review of the diverse field of Artificial Intelligence is also attempted, again with emphasis on a few particular examples. The kind of research which is of interest can be seen as "theoretical psychology". The relevance of Artificial Intelligence to this thesis is in modelling human activity in an attempt to understand human intelligence. The interest is not in the artifact as such but in the intelligence which it artificialises.

The polarisation of Artificial Intelligence into mechanistic bottom-up and behavioural top-down is avoided only in a few instances; two such examples are given - one from the field of Artificial Intelligence itself, the other from psychology.

5.1.2 Chapter 2

In Chapter 2 the game of Kenju, which is the game the simulated player is learning, is introduced. The reasons for choosing this particular game are given as its inherent simplicity and lack of arbitrary restrictions. Both these properties make for an easy implementation and also avoid any distractions such as investigating the properties of the particular game rather than understanding a learning process.

The rest of Chapter 2 describes the software support for the game and teaching environment. First the graphics and keyboard facilities are described. These allow interaction with the learning algorithm through the game and in terms of internal parameters, and provide for automated experimental trials. The internal representation of the board state is then given.

The robustness of these programs has been one of the successes of this work. It is attributed to the particular choice of abstract game-state representation.

The explanation of computer player algorithms is the subject of Chapter 3.

5.1.3 Chapter 3

The evolution of a computer program through four identifiable versions gives the framework for Chapters 3 and 4. Chapter 3 concentrates on mechanism; Chapter 4 on training and learning behaviour.

This evolution separates into two distinct phases: the initial phase starting with System 1 and evolving into System 2; and a second phase, taking a different tack, with Systems 3 and 4.

The first phase relies on a relatively complex learning element to evaluate each move. Sufficient problems are encountered to discourage development along these lines beyond System 2.

System 3 is a departure from this emphasis on acquiring complex knowledge. Instead, a more sophisticated mechanism for directing the player's search for a move is used. This models the chain of attention across the board when a player is looking for a good move. It is during this exploration of the game that the player learns. In System 3 he learns by enquiring about line patterns while in System 4 he enquires (more naturally) about moves.

The learning mechanism for Systems 1 and 2 is a RAM network configured as a learning net which is trained to recognise and value patterns in the game. These patterns correspond to the two-dimensional moziac of black, white, and unoccupied places surrounding a move, although this is somewhat obscured by the method used to code these patterns. A short explanation of basic RAM learning net techniques is included in Chapter 3.

The introduction to System 2 includes an hypothesis which was used in an attempt to bring some discipline into the proliferation of system enhancements which culminated in System 2. Rigid adherence to this hypothesis was not maintained, but in retrospect its influence on the rest of the work can be seen. The essence of the hypothesis is the conflict between expanding knowledge and consolidating it: the advantages of fine-tuning the use of existing knowledge, with the consequent focilization of abilities around this current knowledge, and the gamble of shaking up the status-quo in the hope of achieving an improved final outcome when the knowledge base settles down again.

The description of System 3 is largely an explanation of the move search expressed as a flow diagram. A number of search paths are followed until the algorithm "looses patience" and takes the best move found during its exploration of the board. This exploration can be directed toward defensive and/or attacking moves by two parameters (covered more fully in Chapter 4).

System 4 arose out of a dissatifaction with the teacher-learner interaction of System 3. Instead of taking the computationally obvious choice of line-oriented advice to refine the computer's knowledge of useful line patterns, move-oriented advice is chosen in System 4 for its naturalness, at least for the teacher. This also allows more scope for the computer learner to interpret the advice in its own terms, rather than being told exactly what change should be made to its knowledge.

5.1.4 Chapter 4

How each of the Systems, 1 to 4, learn and behave in the game is the subject of Chapter 4.

Early problems with System 1 nicely illustrate the kind of misunderstanding which can occur between a teacher and pupil when they are not on the same wave length. When encouraging situations containing winnable lines the teacher (the author playing as opponent in System 1) thought he was indicating the value of winnable lines. However, the pupil, in the form of the learning algorithm of System 1, saw, not the winable line, but the more numerous vacant places around it, as the object of the teachers recommendation. The resulting "timid" behaviour of the player after such training, apart from being somewhat comical considering the aim of the game, was also a lesson in teaching. It is usually preferable for the pupil to learn from answers to his own questions, thrown up by his own understanding of his knowledge. In this way we can be more confident that the teachers advice will be taken in the way he intends, since he is responding to a request initiated by the pupil finding himself in a quandary. As long as the pupil's question conveys to the teacher the precise nature of his quandary, the teacher can respond with precise advice to resolve it, and the pupil can assimilate the advice.

Conversely, if unsolicited advice is thrust upon the pupil he can do no better than interpret the meaning as best he can in the absence of any perceived need for advice.

The numerous corrective measures taken to improve the mechanism of training during System 2 can be seen as somewhat futile in the light of

this lack of communication between teacher and pupil. However, the techniques of pattern biasing, and those employed to prevent saturation of the learning net, may have application elsewhere. Both System 3 and 4 transfer the initiative for learning to the pupil, in that the learning algorithm requests advice when it would otherwise, on the basis of its present knowledge, have to make an arbitrary decision.

The question asked when advice is requested falls naturally out of the algorithm's search for a move. The path of the search follows the sequence of best move in line leading to best line intersecting at that move, and so on. In System 3 advice is asked of the teacher when, according to current knowledge, two compared lines are valued equal. The teacher is asked to resolve which of these two, vying for best intersecting line, should be taken as such, in extending the search path. Only if the teacher answers that both lines are equally valued does the algorithm choose arbitrarily. Otherwise the teacher's advice indicates which line should be valued higher than the other, and it is this line which is followed to continue the search.

The technique is the same for System 4 except it is a quandary in choosing the best moves which triggers the learning in System 4.

Results presented in Chapter 4 are from experiments with System 3. The first experiments (Performance Graphs) attempted to measure the level of skill attained through learning. It was hoped that learning curves based on this measure of skill could be used to compare the effectiveness of different training schemes. The experiments were abandoned for lack of consistency in these skill measurements. The increased number of trails thought necessary to get adequate consistency is not considered worthwhile.

The bulk of results presented show the effect of training on the content of the line evaluation learning element. Four ways of presenting the same results are tried, none of which reveal any correlation with variations in training.

Despite these disappointing results, considerable understanding of the way the line evaluation element learns has been gained.

Chapter 4 also goes into some analysis of the effect of some of the parameters which control the actions of the playing algorithm. The results given show the expected effects on playing behaviour. This can be seen clearly with the defensive and attacking play induced by the two parameters, DEFENDING and ATTACKING.

5.2 Analysis

The previous section has considered the preceding chapters in detail; what follows here is a higher level discussion of points arising out of the work as a whole. Much of the thesis has been an account of a largely unsuccessful attempt to model aspects of intelligent behaviour on a digital computer. Why, we should ask, have these attempts so often failed? The discussion to follow may throw light on the underlying reasons.

Throughout the development of the four models which have been described one major obstacle has repeatedly presented itself. That obstacle is the disparity between the intelligence being modelled and the vehicle required to support it - the computer. E.W.Kent (1978) confronts the same problem.

> "Why are we having so much trouble making our computers behave in brain-like fashions ... we are trying to make a wrench do a screwdriver's job."

Ordinarily, this disparity would be considered as no more than an accepted constraint of computer modelling. But in this work the form which the model should take was not well defined at the outset as, for example, Colby's simulation of a neurotic is based on a psychological theory of neurosis (in M. Boden 1977). The process of developing the model was itself to be instrumental in defining the final model.

Such a method, with its appearance of the haphazard, may be criticized as a general simulation method. However, in this case the objective

was not a simulation faithful to any given theory of intelligence. Working from the premise that intelligence is an almost superficial attribute of a very complex system which is not well understood and can not therefore be directly modelled, the approach taken was to build a complex system and with luck one might observe intelligent behaviour.

5.2.1 Development by Evolution

The hope was to simulate the character of naturally occurring intelligence by allowing an analogous intelligence to evolve in the context of a developing computer program modelling a learning game player.

The rationale is essentially the same as that of Fogel, Owens and Walsh (1966) in their book "Artificial Intelligence Through Simulated Evolution." In this book they show how finite state automata can develop the ability to recognise sequences of events. Successive generations of automata are mutated by randomly altering their state transition behaviours and selecting for survival according to success at recognition. The developing automata were found to be capable of adapting to the properties of their environment - the sequences - for simple arithmetic and geometric progressions but also for the more subtle prime number sequence.

The potential for detailed and accurate knowledge of the internal workings of any computerised model should facilitate analysis and could be used in making the design choices analogous to natural selection. It is these design decisions which would mould the developing complex system into an intelligent system. This accessability of arbitrarily

detailed knowledge of the intelligent system is very obviously lacking in our studies of intelligence in its natural form. The computer with its known mechanism was seen as a well understood and controlled world in which to develop, or evolve, a degree of intelligent behaviour. The selected game is also well defined and understood, which avoides obscuring the learning process with unhelpful imponderables. It was hoped it would be possible to trace the evolutionary causes of any intelligent characteristics, and thus to understand their mechanisms.

Two possible misunderstandings must be cleared up here. It is important to understand the distinction between avoiding unknown complexities and requiring a complex model to exhibit intelligence. Any developing model must not simply exhibit intelligence but should provide an example of intelligence which can be analysed and understood. (We know how to produce beings which exhibit "hidden" intelligence.)

This degree of definition and knowledge is important for the programmer but it should not be made available (as built-in knowledge) to the model which is to evolve. A possible theory of intelligence is that it evolves to allow beings to cope in an unknown and ill-defined world (Uhr (1973) refers to ill-formed problems). Thus the goals of the game are nowhere programmed into the computer player. He sees the game as a riddle, and must develop intelligence if he is to learn the game and thus survive as a successful model.

The point is that although the definition of the game is unknown to the programmed player, it must be well understood by the programmer who must analyse the results and act upon them.

5.2.2 Nature and Computing Compared

It was found that although the computer programming environment may be a suitable environment for supporting evolution, it represents a very different environment from that in which natural intelligence may be assumed to have developed. E.W. Kent again:

> "The brain and the computer have both developed in an evolutionary manner, with "survival of the fittest" determining what features were retained and what were discarded. The differences in their designs arise from the fact that nature and computer engineers have different notions of what constitutes "fittest"."

It is this difference which is considered to be the underlying problem accounting for the lack of success in evolving a computer model of intelligent game playing behaviour and learning.

If "computing" and "nature" are compared as environments it can be seen that they differ in very fundamental ways, which it is argued, tends to direct any evolution within them in correspondingly different directions.

"The brain never had speed on its side ... but it never lacked quantity."

"In the solution of mathematical and logical problems, it frequently doesn't make sense to attempt simultaneous solutions of parts of the problem. The results of one operation are essential to the beginning of the next." - i.e. serial computation.

"The brain and the computer have each found a design best suited to the problems they are required to solve and the hardware available, albeit big brains can do some computer-like functions, poorly; and big computers can do some brain-like functions, poorly."

E.W.Kent

Historically, cost factors in computer design (and computing in general) have pushed designs towards the use of a few multi-purpose elements which are shared between their many uses. Hence we have the single general-purpose central processor through which all (and various) computations pass. We have the general storage resources of registers and main/backing store which are less limited in number but still general purpose in that all storage media can (and often do) store all types of data. Similarly, software tends to be general purpose where possible.

There is no identifiable "memory unit" in the brain - the capability for memory is probably integrated wherever it is needed throughout the

Central Nervous System. In general nature integrates functions on a distributed basis, near where they are needed, whereas computers centralise functions in distinct units.

Nature seems to indulge to a greater extent in multiples of more specialised elements. For example, countless nerve cells dedicated to communicating between specific points (c.f. the sharing of bandwidth on a computer bus); countless others, each specialised for a particular specific computation; and yet more for sensory input or for motor output.

Individual nerve cells have the capability for a much greater degree of connectivity and parallelism than currently exists in computers and are normally asynchronous devices. They use a system giving a two dimensional signalling medium of place coding (spacial dimension) coupled with frequency coding (temporal dimension). This is very different from the coding schemes of computers.

Natural selection operates gradually through many stages of procreation compared with computer program development which involves comparitively few generations, each of relatively drastic change. The programmer seldom relies on trial and error, with multitudes of experimental "species" - he prefers to concentrate on one or two well chosen approaches. Because the brain has an evolutionary order of development to its structure the major functions have representations at all levels. Computers, however, are not expected to incorporate pieces of older models.

"Rather than eliminating the older structures and duplicating their functions, the newer ones simply took control of the older ones and used them as subprocessors. A fairly general principle of organisation evolved in which the higher level structures control the lower not by turning them on when needed, but by inhibiting their actions except as desired. The beauty of this system is that if a higher centre is suddenly damaged, the older, more primitive units which it normally holds in inhibition are released to function on their own. Thus, damage tends not to eliminate vital functions, but only to downgrade the complexity with which the job can be performed."

E.W. Kent

Many more instances of the fundamental difference between the natural and computer environments can be cited to show that very different forces of selection apply in each and hence very different "beings" may be expected to evolve. Indeed the computer environment may not be conducive to the development of intelligence at all. However, it may equally be the case that the computer environment, if allowed to develop within its own "natural" constraints, would evolve its own equivalent of Man's intelligence. If this were the case, the above differences suggest that a very different intelligence could be expected to evolve to cope with the computer's world.

5.2.3 Motivation

A major philosophical problem encountered in emulating natural intelligent behaviour on a computer is the fundamental lack of any underlying motivations on the part of the computer program. Thus the simulation gives no satisfactory sense of motivation which derives from the driving criterion of evolution - survival. The inclusion of an arbitrary ultimate goal of the form "maximise X" is unconvincing. The program has no believeable "interest" in maximising X, even when X characterises the purpose of the program (e.g. X = win the game), not even if X is the criterion driving the program evolution through selection (i.e. only algorithms which win games survive).

A theory of G. Pask (1960) provides a possible source of "motivation" which is believeable. Pask basically attributes the urge to do things to the availability of spare capability. An intelligent being has the capacity to do a certain amount of thinking. If his environment does not provide sufficient tasks to fully employ this thinking power, the surplus potential for thought will be soaked up in finding new tasks - "the devil finds work for idle hands to do." This is essentially a theory of boredom as a negative source of motivation or drive.

An equivalent point of view is held by G. Kelly (1955 & 1963) in his theory of personality "The Psychology of Personal Constructs". He was concerned to construct a theory of personality which obviates the necessity of explaining what prods an organism into action.

> "The being we start with is already alive and kicking - we do not have to invent anything to "motivate" him!".

Another "believeable" approach to motivation formulated in terms of goals is the driving criteria used in the work done at Warwick University with industrial robots [M. Larcombe 1979]. The Warwick robot is a mobile robot intended to perform such tasks as carrying goods from place to place in an automated warehouse. Hence the robot is based on a converted fork lift truck.

The robot is given a sequence of tasks involving picking goods up at a receiving bay, say, and carrying them to appropriate pre-defined shelves where they are stocked. The algorithm allows the robot to navigate around the warehouse in the presence of known obstacles such as shelves and unexpected obstacles such as dropped boxes and even other robots or manually operated trucks. The robot program therefore has a semi-static internal model of the warehouse developed from a pre-programmed initial layout.

The navigation part of each task is expressed as a goal position, to be reached by a possibly complex route found by exploration. The destination is represented as an attractive force and obstacles as repellant. Taking their lead from physics the Warwick workers give these forces an inverse square law. The significant innovation with respect to motivation is their algorithm, or heuristic, to explore alternative routes and for backing out of dead end passages.

They make an analogy with an animal who leaves behind a trail of droppings whose odour is repellant to it. The animal will normally be attracted in the direction of its goal, but if this leads it down a blind alley, eventually the stench of its own droppings will drive it out of the cul-de-sac. A refinement whereby the odour of its droppings

decays with time allows the animal (or the robot) to back-track and search out an alternative route to its goal.

It can now be seen how the robot's strategy is similar in effect to Pask's "fidget" theory (above). The robot cannot help his smelly droppings but he can move around in an attempt to get away from the stench.

A generalisation of the robot's strategy can more faithfully reflect real world environments. The single, well defined, positive goal in the presence of pre-defined, and also unexpected, negative anti-goals can be extended to a more normal situation in which multiple, known and unknown, positive goals and negative anti-goals exist.

What is "known" and "unknown" usually varies with time and position in the environment. For instance the robot might be attracted by lights (among other positive goals). Now, if in its exploration of the warehouse the robot rounds a corner, a previously obscured source of light might come into its field of view. If this light is sufficiently "bright" it could divert the robot's attention by overriding its previous goals. This is simply a positive example of what happens with a build up of negative foot odour - the temporal strengths of forces causes the robot to pursue dynamically changing goals.

This senario of dynamically changing goals has been developed from three simple and believeable premises:

i) that an intelligent mechanism would have an inherent capacity for activity (i.e. state changes) which would not vary to accommodate the demands made on it.

ii) that the mechanism's interaction with its environment would be such that any excess capacity for state changes would tend dynamically to reveal new areas of the mechanism's state space and obscure old ones.

iii) that the effect of evolutionary processes on the developing mechanism would be to make goals of situations which lead to success (in terms of survival) and anti-goals of those leading to failure.

A somewhat contrived illustration might be that of an animal employing its capacity for activity in grazing. The result is a lack of grass in its immediate vicinity. This leads the animal away from the vicinity to new grass which in turn takes it toward water; whereupon it forgets its hunger in favour of a previously forgotten thirst.

This view of "local temporary goals" contrasts with the usual hierarchy of goals in which subgoals are elaborations of higher goals (which is

probably an over rationalised interpretation encouraged by computer implementations).

The importance of locality in the scope of goals is recognised by Colby (in M. Boden 1977) where the goal to reduce conflict applies only to conflict between beliefs in a localised pool of currently relevant beliefs (a "complex" in Colby's terminology). Similarly in Kelly (1955 & 1963)

> "When one limits the realm of facts, it is possible to develop a detailed system without worrying about the inconsistencies in the system which certain peripheral facts would reveal."

5.3 Summing Up

Much of the effort in Artificial Intelligence has concentrated too much on producing an artifact having forgotten about intelligence. The scientific curiosity which originally inspires work in Artificial Intelligence is so easily left unsatisfied by the engineering approach where programming becomes all important and the task becomes one of efficient data base management. Much of this abstracted data base research concludes having offered no suggestion as to how such organisations of information might be supported (or might have evolved) in the human mind - the subject of their curiosity.

This is why so much of Artificial Intelligence understands the characteristics of the information but fails to explain the intelligence which generates and manipulates it. The myriad techniques of tree-building and searching, with or without backtracking, are more

techniques of programming than the mechanisms of thought. N.J. Nilsson's (1980) book "Principles of Artificial Intelligence" is entirely devoted to techniques of building and searching such data structures.

Game playing itself is often chosen for its similarity with methematical problem solving - one of the least typical of intelligent pursuits, being unusually well defined and "mechanical". Logical deductive reasoning, which is the basis of most games, is only part of what interests us about intelligence. Viewed on its own, it lacks an essential aspect of intelligence - that of coping in the absence of a logical course of action. This is the basis for distinguishing ourselves from the less or unintelligent species. They are so logical that they cannot think.

A view of intelligence, or its origin, which was taken earlier (see section 5.2.1) is that intelligence develops to cope with an ill-defined and/or changing world. This suggests that intelligence only develops in inhabitants of partially understood worlds whose laws (or rules, in the case of a game) are unknown, or only partially known and understood by these inhabitants. The real world is more of a puzzle than a game.

Whereas this state of affairs might also exist in a computer context (e.g. the model developed in this thesis) it should be borne in mind that computing has grown up in response to the need to find solutions to well defined (and therefore known) problems. The computing structures which have developed are tailored to precisely describing and solving known and pre-defined problems. Hence the computer environment as we know it, may be incapable of evolving intelligent inhabitants if the above view of intelligence is appropriate. Indeed, the pursuit of

intelligence may be premature in the context of such an early evolutionary epoch as that which computing has reached.

It is interesting to note that current trends in distributed computer architectures and software methodology are lessening the differences between computers and natural phenomena such as brains and social structures.

Further pursuit of this line of thought is a possible direction for further study. If the evolution of an alternative computer-oriented (rather than natural) intelligence were to be achieved, it should provide many answers to questions about intelligence itself. Many of the red herrings encountered in Artificial Intelligence research may well be due to the lack of a generally agreed understanding of what is meant by "intelligence". Given more examples than the single example currently available, a comparison of alternative intelligences should clarify what is, and what is not, intelligence. A better understanding of what we mean by intelligence would be a very worthwhile outcome of such a piece of research, quite apart from the achievement of evolving an alternative intelligence in itself.

REFERENCES

Aleksander, I.

"Microcircuit Learning Computers."

M & B Monograph EE/4, 1971 ISBN:0.263.51724.1

Aleksander, I., Stonham, T.J. & Wilson, M.J.D. (1974)

"Adaptive Logic for Artificially Intelligent Systems."

The Radio and Electronic Engineer, Vol.44 No.1 Jan 1974.

Andreae, J.H. (1977)

"Thinking with Teachable Machine."

ACADEMIC PRESS, 1977 ISBN:012.060050.1

Bannister, D. and Fransella, Fay (1971)

"Inquiring Man: The Theory of Personal Constructs."

Penquin Books, 1971 ISBN:0.14.08.0617.2

Boden, Margaret A. (1977)

"Artificial Intelligence and Natural Man."

The Harvester Press, 1977 ISBN:0.85527.435.2

Bond, A.H. (1973)

"A Bibliography of Game Playing by Computer."

Queen Mary College, Computer Science & Stats. Dept. 1973

Elcock, E.W. & Murray, A.M. (1967)

"Experiments with a learning component in a Go-Moku playing program."

in Machine Intelligence 1, eds. Collins, N.L. & Michie, D.

Elcock, E.W. & Murray, A.M. (1968)
"Automatic description and recognition of board patterns in Go-Moku."
in Machine Intelligence 2, eds. Dale, E. & Michie, D.

Feigenbaum, E. & Feldman, J. eds. (1963)
"Computers and Thought."
McGraw-Hill, 1963

Fogel, L.J., Owens, A.J. & Walsh, M.J. (1966)
"Artificial Intelligence Through Simulated Evolution."
Wiley, 1966

Kelly, G.A. (1955 & 1963)
"The Psychology of Personal Constructs." Vols. 1 & 2
W.W. Norton, 1955, reprinted 1963

Kent, E.W. (1978)
"The Brains of Men and Machines."
4 part article in BYTE Magazine January-April 1978
and BYTE BOOKS/McGraw-Hill ISBN:0.07.034123.0

Koffman, E.B. (1968)
"Learning Games Through Pattern Recognition."
IEEE Trans. Systems Science & Cybernetics SSC-4 No.1 1968

Larcombe, M.H.E. (1979)
"Mobile Robots for Industrial Use."
The Industrial Robot, Vol.6 No.2 June 1979

Michie, D. (1980)

"Expert Systems."

The Computer Journal, Vol.23 No.4

Nilsson, N.J. (1980)

"Principles of Artificial Intelligence."

TIOGA Publishing Co./McGraw-Hill 1980

Oda, M. & Womack, B.F. (1972)

"Special Multimodality and Adaptivity in Renju Game."

IEEE Trans. Systems, Man, & Cybernetics Jan 1972

Pask, G. (1960)

"The Simulation of Learning and Decision-making Behaviour."

in Aspects of The Theory of Artificial Intelligence 1960

Pitrat, J. (1968)

"Realisation of a General Game-playing Program."

Proc. IFIP Cong. 1968

Samuel, A.L. (1959)

"Some Studies in Machine Learning using the Game of Checkers."

IBM J. Res. and Dev. 3, 1959

and in Feigenbaum & Feldman (1963)

Schank, R.C. & Colby, K.M. eds. (1973)

"Computer Models of Thought and Language."

Freeman, 1973

Slagle, J.R. & Dixon, J.K. (1969)

"Experiments with some programs that search game trees."

JACM 16, 1969

Slagle, J.R. (1971)

"Artificial Intelligence: The Heuristic Programming Approach."

McGraw-Hill 1971

Stonham, T.J. & Aleksander, I. (1974)

"Optimisation of Digital Learning Networks when applied to

Pattern Recognition of Mass Spectra."

IEEE Electronics Letters Vol. 10 1974

Uhr, L. (1973)

"Pattern Recognition, Learning, and Thought."

Prentice-Hall 1973 ISBN:013.654095.3

Zobrist, A.L. (1969)

A Model of Visual Organisation for the Game of GO."

AFIPS Conf. Proc.,34,1969

APPENDIX A

Basic Strategies of Kenju

The basic winning strategy for both players is to form two (or more) threats with a single move, as the opponent can usually block only one of these threats with his next move and thus there is a possibility of developing a winning five with the remaining threat.

The basic winning strategies can be summarised as the formation of one of the following, where the terms used are defined in Oda and Womack (1972):

Open Four - If a player succeeds in forming an open four, he will usually win the game, as his opponent can block the open four only on one side with his next move, and the player can form a five (complete a winning five) by adding his stone on the other side with his next move.

4-3 - If a player succeeds in forming a double threat of 4-3, he almost (i.e. not always) wins the game, as the opponent can block only the four with this next move and the player can often develop the threat of the open three into an open four.

4-4 - If a player succeeds in forming a double threat of 4-4, he wins the game because the opponent can block only one threat with his next move and the player can develop the other four into a winning five with his next move.

3-3 - If a player succeeds in forming a double threat of 3-3, he almost (i.e. not always) wins the game, as he can often develop one of the double open threes into an open four, which leads to his win.

APPENDIX B

Command Mode Commands

Commands can be entered during command mode. They can appear one on a line or several on a line separated by ";".

X - exit from Kenju

R - return to game-playing mode

C - change I/O streams

O - output state of board

T - test for win in game

SDn - set debug level to n

SCn - select response conbination routine n

SSn - select address selection mapping n

DP - display the placeblock specified

DPE - display entire placetable

DPT - display a trace of lines containing specified place

DPL - display placetable then linelist

DL - display lineblock specified

DLE - display entire linelist

DLT - display trace of places in specified line

DLP - display linelist then placetable

DM - display table of moves

DS - display space used and unused

DB - display board state

DG - display responses (goodness) for useful moves

DC - display situation codes for useful moves

DR - display net (RAM) contents

I - initialise net contents from I/O stream OLD

E - save experience (net contents) on I/O stream NEW

B - force execution of Blackmove

W - force execution of Whitemove

M - force execution of Makeplace

K - force execution of Killine

L - force execution of Line (line making routine)

F - force execution of Findplace

Pn - play a game using mode n

APPENDIX C

Situation Codes

The first section gives those patterns which correspond to each PL.PM (or OL.OM) sub-code. "X" gives the position of Player (or Opponent) stones.

The second section then gives all possible [PL.PM/OL.OM] combinations.

```
|_|_|_|_|?|_|_|_|_|        0.5

|X        ?|
                          1.1
          |?        X|

|_|X      ?|_|
                          1.2
        |_|?      X|_|

|_|_|X   ?|_|_|
                          1.3
      |_|_|?   X|_|_|

|_|_|_|X ?|_|_|_|
                          1.4
    |_|_|_|? X|_|_|_|
```

```
|X X     ?|
        |?     X X|
|X   X   ?|
        |?   X   X|
|X     X ?|
  |X     ? X|
    |X   ?   X|
      |X ?     X|
        |? X     X|
```

2.1

```
| |X X   ?| |
      | |?   X X| |
| |X   X ?| |
  | |X   ? X| |
    | |X ?   X| |
      | |? X   X| |
```

2.2

```
| | |X X ?| | |
  | | |X ? X| | |
    | | |? X X| | |
```

2.3

```
|X X X   ?|
        |?   X X X|
|X X   X ?|
  |X X   ? X|
    |X X ?   X|
      |X ? X   X|
        |? X X   X|
|X   X X ?|
  |X   X ? X|
    |X   ? X X|
      |X ?   X X|
        |? X   X X|
```

3.1

```
|  |X X X ?|  |
  |  |X X ? X|  |
    |  |X ? X X|  |
      |  |? X X X|  |
```

3.2

```
|X X X X ?|
  |X X X ? X|
    |X X ? X X|
      |X ? X X X|
        |? X X X X|
```

4.1

All possible [PL.PM/OL.OM] combinations:

[0.5/0.5]	[1.1/0.5]	[2.1/0.5]	[3.1/0.5]	[4.1/0.5]
[0.5/1.1]	[1.1/1.1]	[2.1/1.1]	[3.1/1.1]	[4.1/1.1]
[0.5/1.2]	[1.1/1.2]	[2.1/1.2]	[3.1/1.2]	[4.1/2.1]
[0.5/1.3]	[1.1/1.3]	[2.1/1.3]	[3.1/2.1]	[4.1/3.1]
[0.5/1.4]	[1.1/1.4]	[2.1/2.1]	[3.1/2.2]	[4.1/4.1]
[0.5/2.1]	[1.1/2.1]	[2.1/2.2]	[3.1/3.1]	
[0.5/2.2]	[1.1/2.2]	[2.1/2.3]	[3.1/3.2]	
[0.5/2.3]	[1.1/2.3]	[2.1/3.1]	[3.1/4.1]	
[0.5/3.1]	[1.1/3.1]	[2.1/3.2]		
[0.5/3.2]	[1.1/3.2]	[2.1/4.1]	[3.2/0.5]	
[0.5/4.1]	[1.1/4.1]		[3.2/1.1]	
		[2.2/0.5]	[3.2/2.1]	
	[1.2/0.5]	[2.2/1.1]	[3.2/3.1]	
	[1.2/1.1]	[2.2/1.2]		
	[1.2/1.2]	[2.2/2.1]		
	[1.2/1.3]	[2.2/2.2]		
	[1.2/2.1]	[2.2/3.1]		
	[1.2/2.2]			
	[1.2/3.1]	[2.3/0.5]		
		[2.3/1.1]		
	[1.3/0.5]	[2.3/2.1]		
	[1.3/1.1]			
	[1.3/1.2]			
	[1.3/2.1]			
	[1.4/0.5]			
	[1.4/1.1]			