To the Graduate Council:

I am submitting herewith a thesis written by Lawrence R. Webb entitled "Game Stage Classification Using Machine Learning and Fuzzy Sets for the Game of Othello". I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Maria Zemankova

Maria Zemankova, Major Professor

We have read this thesis
and recommend its acceptance:

David C. Rine

Accepted for the Council:

Vice Provost
and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in her absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature 

Date 7/25/88

Game Stage Classification Using Machine Learning and Fuzzy Sets for the Game of Othello

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Lawrence R. Webb

August 1988

ABSTRACT

Can the performance of a program to play Othello be improved using machine learning techniques? In this research two paths of research were pursued to attempt to answer this question.

In the first method, a program playing an uninformed strategy competed against a program playing a fully informed weighted square strategy. Locations occupied by the winner of each game were used to modify the weightings of the uninformed strategy. After a series of games the uninformed strategy was able to compete favorably against the expert strategy. This showed improvement in the performance of the learning program when competing against an informed strategy. Low and high thresholds of acceptance for location weightings were used showing that the high threshold produced a better playing program.

Once it was shown that machine learning can be used to improve the performance of an Othello playing program, the machine learning technique was implemented for a second testing method. By identifying characteristics of opening, middle and end game situations, game board configurations at each turn in the game were classified into these sets using fuzzy set theory. Different strategies at each stage of the game were utilized to maximize the performance of the program at each of these stages.The parameters of these fuzzy sets were automatically modified by the program at the end of each game. By adjusting the point in the game when the opening game strategy stopped playing and the middle game strategy began, and by adjusting the point in the game when the middle game strategy stopped playing and the end game strategy began, an improved strategy was defined.

This study shows that the performance of a program playing Othello can be improved using even simple machine learning techniques. Combining fuzzy set theory with machine learning to solve game theory problems is also

shown to be an effective technique to improve the quality of play when exact quantifications of characteristics of a game are uncertain.

Contents

1 INTRODUCTION **1**

1.1 Problem Definition 1

1.2 Methodology 2

1.3 History of Implementation 4

1.4 Overview of Thesis 6

2 GAME RESEARCH **8**

2.1 Game Theory 8

2.2 Othello as a Zero-Sum Game 10

2.3 Fuzzy Sets Applied to Othello 12

2.4 Machine Learning in Game Theory 14

3 DESCRIPTION OF OTHELLO **18**

3.1 Termini Technici 18

3.2 History and Rules of Othello 19

4 USING WEIGHTED SQUARES IN OTHELLO **26**

4.1 Strategies to Play Othello 26

4.2 Methods of Improving the Play 31

4.3 Modifying the Weighted Squares 32

4.4 Weighted Square Strategy Conclusions 38

5 CLASSIFICATION OF GAME STAGES 42

5.1 Definition of the Fuzzy Functions 42

5.2 Fuzzy Components for Othello 46

5.3 Combination of Components 51

5.4 Description of Strategies for Each Stage 55

5.5 Modification of Weighting Coefficients 57

5.6 Modification of Fuzzy Set Parameters 59

5.7 Using Machine Learning to Modify the Parameters 62

5.8 Conclusions of Game Stage Modification 68

6 CONCLUSIONS 71

6.1 Success of Research 71

6.2 Application of Research Outside Game Theory 73

6.3 Areas of Continued Research 75

BIBLIOGRAPHY 79

VITA 82

List of Figures

3.1 Othello Starting Configuration 19
3.2 Example 1–Before 21
3.3 Example 2–After 21
3.4 Example 2–Before 22
3.5 Example 2–After 22
3.6 Example 3–Before 23
3.7 Example 3–After 23

4.1 Invicta's weighted square strategy 29
4.2 Arnold's weighted square strategy 29
4.3 Disc differentials 30
4.4 Disc difference with low threshold of acceptance. 35
4.5 Final student weightings with low threshold of acceptance. 36
4.6 Equivalent disks. 36
4.7 Disc difference with high threshold of acceptance and mutation. . 39
4.8 Final weightings with high threshold of acceptance. 40

5.1 Ascending S-graph for the set *crowded.* 45
5.2 Descending S-Graph for the set *not crowded.* 47
5.3 Bell S-Graph for the set *comfortably crowded.* 48
5.4 Regions for stage classification 50
5.5 Degree of memberships in Move Sets 52

5.6 Degree of memberships in Game Stage Sets 56
5.7 Degree of memberships in game stage sets with modified parameters 61
5.8 Disc differences between original strategy and new strategy . . . 63
5.9 Game stage sets with modified start moves parameters 65
5.10 Game stage sets with modified late moves parameters 67
5.11 Degree of memberships in Game Stage Sets using Dilation 69

Chapter 1

INTRODUCTION

This thesis uses machine learning to improve the quality of a program playing the game of Othello. A technique combining machine learning and fuzzy set theory will be used to assist in improving the quality of the game playing.

1.1 Problem Definition

There are two questions that this thesis will attempt to answer. The first is: Can a simple machine learning strategy improve play based entirely upon final disc count and location? There are many complex strategies that have been proposed to play the game of Othello. This paper will investigate a strategy known as a "weighted square" strategy. With this strategy, different locations on the playing field are ascertained to be more advantageous to control. A machine learning program begins by being completely uninformed about any location advantages. This program is expected to be able to discern these locations through repeated play with an informed strategy. If the program playing with the uninformed strategy is in fact learning, then its play should improve against the program playing the informed strategy.

The second question this paper will attempt to answer is: Can the criteria of a "won game" used to improve the play for weighted square also work

to improve play using more complex strategies? To answer this, the machine learning strategy will not concentrate on improving one strategy, but instead will employ three strategies. Each of these strategies will be used at different stages of the game. By modifying when each of these stages of the game occur, the machine learning program should improve its play based upon the outcome of the game.

Since there are no clear distinctions between the stages within an individual game, heuristics must be chosen with which to distinguish the stages. By modifying the weighting and parameters of these heuristics, the stage of the game that each individual board configuration is a part of and thus the strategy that is used to determine the best move is also modified.

The technique of using machine learning to modify fuzzy set parameters is applicable beyond the game playing environment. If the automatic modification of the game stages works as expected, then this technique could be applied to a wide variety of problems where some feedback is provided to compare the eventual quality of the classification.

1.2 Methodology

To explore machine learning, the first task was to develop a strategy that played reasonably well and was able to improve itself. The strategy first chosen was a weighted square strategy. The research would pursue a method of automatically modifying an uniformed weighted square strategy to compete effectively with an informed weighted square strategy. In this phase of the research, the informed strategy is a strategy that some experts (and preliminary stages of this research) have found plays reasonably well. The uninformed strategy plays what can be could be called a *novice* game, or an almost random game.

To improve the play of the uninformed strategy, two computerized op-

ponents will compete. The first task to achieve this was to set up a monitor program to keep control of the current board positions and which player's turn it is. The code to take care of this aspect of the game is in the main module called OTY . This module always allows X to go first and keeps a timer on each player as they move. The timer is for statistical information to show how much time was used for each move. There are no constraints on the amount of time a player is allowed, but generally search is tailored to depths so that each player would use no more then 45 minutes of total time (90 minutes for a full game).

One computerized player will be using an informed weighted square strategy. The other computerized player will be the learning program. The learning program will begin with a fully uninformed weighted square strategy. As each game is played, the learning program will modify it's strategy in an effort to improve its play in the subsequent game. It must be emphasized that this research is not designed to create a championship Othello playing program. It is only designed to show improved play.

Once this is completed, the research will take a different approach. Instead of using one strategy and making changes to that strategy to improve the computer's play, a program would be developed that would employ three simple strategies. These strategies are implemented at different stages during the course of a game. By using fuzzy set theory, the stage of the game could be determined. Machine learning would be shown to be an effective tool in this approach by modifying the heuristics determining when in the game the different strategies are used.

Describing the stage of the game, opening, middle game or end game, implied the use of certain heuristics specific to Othello. Since game board positions needed to be classified into sets based on parameters, these parameters could be modified by the learning program. Once again, the goal of this research is only

to show improved play. If the play of the Othello game can be improved, then this technique could be used in other areas where classification of a changing situation may be useful and the parameters of that classification are uncertain.

Success of both of these methods of machine learning was based solely upon improved play. This researcher makes no claim as to the validity of the strategies compared with other strategies. The strategies are simply a means of demonstrating the techniques involved in machine learning.

1.3 History of Implementation

The original implementation of Othello by this researcher was undertaken to fulfill the requirements of a graduate class in data structures. In the original implementation, a mini-max depth-first search was used to search for a move that produced the most disks for the computer. It was written on the University of Tennessee's VAX 785 entirely in Pascal. The computer controlled which player's move it was. The human opponent was able to enter his move via the keyboard. The program played well enough that it was initially able to beat myself, the person who had designed and written it.

Other graduate students became interested in the computerized game. Some started playing it more often and soon realized the maximum disk heuristic that the computerized strategy employed. The human opponents were able to use this to their advantage to beat the computer program. Even beginning players recognized that some locations on the board are better to occupy than others.

In an effort to improve the game playing program's performance, this researcher chose to increase the search depth. To increase the search depth without making the search times prohibitively long, the program was rewritten in "C". As it was rewritten, $\alpha - \beta$ pruning was added to the code performing

the move search. No attempt was made to order the moves being examined by any heuristic. In experiments for this thesis, $\alpha - \beta$ pruning reduced by about 50 percent the amount of time that was required for each additional search level. Moves were evaluated in sequential order from the upper-left corner to the lower-right. Since no attempt was made to prioritize the order in which the moves were evaluated, the $\alpha - \beta$ pruning did not make as significant an improvement as may have been possible. But ordering the sequence may not have been practical. If the best paths were known at the beginning then the search would not be necessary.

> The effectiveness of the alpha-beta procedure depends greatly on the order in which paths are examined. If the worst paths are examined first, then no cutoffs at all will occur. But, of course, if the best path were known in advance so that it could be guaranteed to be examined first, we would not need to bother with the search process. ... A doubling of the depth to which the search can be pursued is a significant gain. Even though all of this improvement cannot typically be realized, the alpha-beta technique is a significant improvement to the mini-max search procedure[15].

Search depths of 6 and 7 turns using $\alpha - \beta$ pruning were no longer prohibitive. Contrary to what this researcher had expected, doubling the search depth did not significantly improve the quality of play even though the search strategy was still based on a maximum disk strategy. Rosenbloom observed the same situation at a tournament in which his program competed. "In the Northwestern tournament, the two worst programs searched to an average depth of 7–8 ply, while the better programs searched to only 4 or 5 ply."[16]

After taking additional classes in artificial intelligence, this researcher realized that machine learning could be an effective tool to improve the computerized playing of Othello. Additional research showed that Othello meets all of the requirements of a good zero-sum game and only some research has been done on the game of Othello. Because of these facts, this researcher choose to continue developing Othello for this thesis.

Originally the program was implemented on a VAX but this researcher felt that moving the research to a minicomputer may provide a more constant environment in which to compare search times. Since the program had been rewritten in C, moving to an IBM/PC-AT clone system required only minor programming changes [1]. The compiler used on the AT is the Borland Turbo-C.

1.4 Overview of Thesis

Chapter 2 of this thesis presents a brief overview of game theory, specifically zero-sum games. Work being done with Othello is emphasized. This chapter also includes some descriptions of work currently being done in the areas of machine-learning in game playing programs. Included in this is not only Othello, but also some of the more frequently researched games such as checkers. A discussion of fuzzy sets concludes Chapter 2.

Chapter 3 covers the specific rules concerning the game of Othello. Examples are given of moves and some discussion is given to why certain strategies are researched more then others.

In Chapter 4 the results of research into using machine learning to modify a "weighted square" strategy are presented. It shows results of using different specifications on how the program actually learns.

Chapter 5 describes the heuristics used to classify game stages. It also describes the techniques used to to modify these heuristics. Also in this chapter, the strategies used at each of the game stages are described. The results of these machine learning programs are presented as proof that this technique can be used to improve play.

In Chapter 6, a short discussion is presented proposing possible modifi-

[1] Integers in Turbo-C and in VAX C are of a different length. This meant that some of the macros used in the program had to be modified to reflect this fact.

cations to the classification strategy to meet other needs. It becomes obvious as the research is presented, that many of the heuristics to classify the game stages are specific to Othello. By using heuristics specific to a problem, the machine learning technique shown in this thesis could be used to improve different types of classification not only in a game environment but also in other areas where the parameters of a fuzzy set are uncertain.

Chapter 2

GAME RESEARCH

2.1 Game Theory

Zero-sum games have long been of interest to researchers in the area of game theory. Charles Babbage considered using his Analytical Engine to play chess and later to play Tic-Tac-Toe [4]. The first successful attempt using a computer to play a game was by A. Samuel in 1963 playing the game of checkers [17]. Since the rules and characteristics governing n-person games where $n > 2$ are not as clearly defined, this research will only deal in the area of two-person games.

The idea of gaming and competition has been studied since 1944 and John Von Neumann's *Theory of Games and Economic Behavior.* To define what a game is is not difficult [21].

> If two or more persons exchange goods with each other, then the result for each one will depend in general not merely upon his own actions but on those of others as well. Thus each participant attempts to maximize a function (his above-mentioned "result") of which he does not control all variables. This is certainly no maximum problem, but a peculiar and disconcerting mixture of several conflicting maximum problems. Every participant is guided by another principle and neither determines all variables which affect his interest.

Two-person, zero-sum games are important to researchers because they provide a structured task in which it is very easy to measure success or failure[15]. It is also important from the standpoint of advanced research to have a game

that is non-deterministic in the practical sense. Given an infinite amount of time and computer resources, every possible outcome from any game situation in games such as chess or checkers or Othello could be explored. This would be impossible, though, in a practical sense. For chess, an exhaustive search may entail as many as 10^{120} moves which have to be evaluated[5]. Assuming that 1000 configurations could be evaluated every second, the time required to make one move would be $3.1 * 10^{109}$ years! Since an exhaustive search is impractical, it is clear that heuristics thus become important in determining the chosen moves[15].

A discussion of the characteristics of a zero-sum game is important. Zero-sum games are used in many areas of research. Amos Tversky works primarily in the field of mathematical psychology but has done much work in the area of game playing. Tversky defines a two-person, zero-sum game in the following manner[6]:

> Strictly competitive games are games where the interests of the two players are strictly opposed. Hence, the utility of player R is the negative of the utility of player C for all outcomes, ...such games are also called zero-sum games.

An example of a zero-sum game would be a simple game of marbles. To play marbles, a circle is drawn of any diameter (usually about 2 feet) on a flat sandy surface. All of the marbles of all of the players are placed within the circle with the exception of each player's "shooter" marble. Each player alternately attempts to roll his shooter marble from outside the circle in an attempt to knock marbles within the circle out of the circle. The player is allowed to keep any marbles that he knocks out of the circle. He then retrieves his shooter marble and it becomes the next player's turn. Play continues until all of the marbles have been knocked out of the circle.

For our example we will consider a game of marbles with two players. Each of the two players brings a finite number of marbles, for example twenty.

Each player is not allowed to leave the game to bring more marbles into the game or to remove marbles from the game. This will lead to a finite and countable number of marbles in the game, in this example 40. As the play proceeds, the actual number of marbles that each player controls will change. It is obvious here that if a player wins control of an additional marble then an opponent forfeits control of that marble. The total number of marbles will stay constant at 40 but the number of marbles controlled by each player will change.

In this example, as the number of marbles controlled by one player increases the number of marbles his opponent may control is directly and equally decreased. The sum of the increases and decreases is zero. When all of the marbles are knocked out of the circle, the game is over. The player with the greater marble count at that time is declared to be the winner.

In the case of a two-person zero-sum game, each gain by a player is directly felt by the other player. Using zero-sum games is common for computer research because, based on the rules for determination of a winner, a machine's play for an individual game can clearly be called more effective if it increases the score and less effective if it decreases the score.

2.2 Othello as a Zero-Sum Game

The characteristics that are required to ensure that a game is adequate for machine learning research are well stated by Samuel[17]:

> ...it is fair to state that [the game must] contain all of the basic characteristics of an intellectual activity in which heuristic procedures and learning processes can play a major role and in which these processes can be evaluated.

Samuel formalized these rules to justify his use of checkers or the use of other zero-sum games such as chess[17].

1. The activity must not be deterministic in the practical sense.

2. A definite goal must exist - the winning of the game - and at least one criterion or intermediate goal must exist which has a bearing on the achievement of the final goal and for which the sign[1] should be known.

3. The rules of the activity must be definite and they should be known.

4. There should be a background of knowledge concerning the activity against which the learning progress can be tested.

5. The activity should be one that is familiar to a substantial body of people so that the behavior of the program can be made understandable to them.

There are many examples of zero-sum games that meet the above criteria. The game most commonly used in research is chess. Samuel, on the other hand, focused on checkers because "Checkers, rather than chess was chosen because of the simplicity of its rules permits greater emphasis to be placed on learning techniques"[17].

Othello was chosen for the research in this thesis for similar reasons. The rules of Othello are very simple. This allows the program to concentrate on the task at hand, machine learning, as opposed to the myriad of rules governing the individual pieces used in chess.

Othello meets all of Samuel's criteria. The game is obviously not deterministic in a practical sense. To exhaustively search every possible move based on an average of 5 possible moves at any one time could result in the examination of as many as 5^{60} different configurations. Assuming 1000 board configurations are evaluated each second, determination of each move could take $3.1 * 10^{49}$ years.

[1] "Sign" is used to mean some thing quantifiable such as a *piece advantage*. If your piece advantage is positive it means you are ahead. It it is negative, it may indicate that you are loosing.

A definite and clear goal of the game is for a player to have more discs of their color on the board than the opponent's discs at the conclusion of the game. This fact can be counted easily and a winner and loser determined.

Although some minor differences in individual rule specification exists, the rules most generally accepted can be used.[2] This research uses the rules laid out by Invicta Games Corporation purchased along with the actual game[9].

Othello, although not popularly played, does have expert players. In competitions against machines, experts have competed only marginally well. Experts are refusing to play machines due to many programs' expert playing ability[16]. This validates Samuel's fourth criterion concerning testing the program against a background of knowledge.

This researcher's interest in Othello is the fact that it is a game that he learned in his childhood and continued to play. As his interest in computer science grew, he programmed a zero-sum game as a requirement of a graduate course. The simplicity of programming the rules and the quality of play by the first implementation proved encouraging enough that Othello was chosen for use in this thesis.

2.3 Fuzzy Sets Applied to Othello

An area not pursued by Rosenbloom provides the impetus for the second stage of this research. Different strategies seem to work better at different stages of the game. The determination of game stage is never clearly defined[16]:

> As with many other games, Othello games can be temporally subdivided into three phases: (1) the opening game; (2) the middle game; and (3) the end game. There are no firm boundaries between the phases. Rather, they are characterized by their different strategic concerns.

[2]Rules in this case should not be confused with strategies. For a further discussion of this see chapter 3.

Rosenbloom uses two methods of evaluating the quality of a current game board. One evaluation is a polynomial [3] and is used for the majority of the game. The second evaluation is a maximum disc strategy and is used only during the final moves of an individual game. He makes his determination of which evaluation to use by the following criteria[16]:

> The decision as to which search to perform is based on empirically determined equivalences in search time. Using [the end game evaluation functions] the search can get deeper than is possible with the normal evaluation because of the smaller time costs of the evaluation functions involved.

As fewer moves are left in the game, the search tree is proportionally limited. At a point governed by the time remaining for the player to determine his move, the program automatically begins to use the end game strategy.

Fuzzy set theory provides an effective tool in solving the problem of game stage determination.

What is a "fuzzy set"? To provide an example of a fuzzy set, we must first provide an example of an exact or *hard* set. If the question were asked *"Is this room crowded?"*, a simple *"yes"* or *"no"* would be expected as the answer. Let us assume that upon inspection by a fire marshal, maximum save occupancy is set at 75 people. For this example we will say that with 75 people it is crowded. At this point the room can be defined for any occupancy as either *is crowded* or *is not crowded.*

Consider the situation where there are 20 people in the room. The room can obviously be classified as *not crowded.* Consider, on the other hand, the situation where there are 74 people in the room. Based on our rules we must then conclude that the room with 74 occupants is also *not crowded* – an inference which is seemingly at odds with physical actuality[3].

[3]This polynomial weights different aspects of the current board configuration. Among these aspects are the number of edge stable discs, internal stable discs and unstable discs.

Instead of classifying *crowded* as a *yes* or *no* response, the response could be quantified. A *grade of membership*, $M(x)$, could be assigned to the occupancy of the room. "Put another way, $M(x)$ embeds the two-valued logic of {0,1} in the continuously valued logic [0,1]."[3] Several *membership functions* will be presented in Chapter 5 of this thesis. These functions will provide the basis for the quantification of the factors determining the game stage for the game of Othello in the thesis.

2.4 Machine Learning in Game Theory

> A learning system (or automaton) can be considered as a system (or automaton) which demonstrates an improvement of performance during its operation from a systematic modification of its structure or parameter values[8].

It is important to distinguish learning from knowledge. Many researchers have used the concept of "book moves". This is knowledge that has been proven to work by human players of games. With this distinction made, this thesis will not explore the "book moves".

Rote learning is the most elementary type of learning. With this technique, all of the board configurations encountered during a game are saved along with their evaluations. Should that board configuration be encountered again in a future game, it would not have to be evaluated again. Also, if a board configuration is encountered at the bottom of a search tree, the evaluation returned would reflect an evaluation deeper than the actual search tree. For instance, if the current search was at a depth of 3 moves and a previous configuration is found that has been evaluated to a depth of 3 moves, then the evaluation that can be returned is reflective of a search of 6 moves. "This can hardly be called a very advanced form of learning; nevertheless, if the program then utilizes the saved time to compute further in depth it will improve with time."[17]

This method obviously leads to an extremely large data base if left unchecked. Methods of generalization and standardization of game boards can reduce the actual number of game boards but this alone is not sufficient. Samuel used a form of *forgetting* which was based on frequency of use. Less frequently used boards are soon forgotten while frequently used boards are retained even if they are not thereafter used for a long period of time.

Current research by Kenneth DeJong and Alan Schultz on Othello uses experience-based learning techniques as a method of improving play. For their research, board configurations and various statistics concerning the board configuration are kept. Information about the mini-max value of the board are kept in the statistics. Actual moves tried are kept as arcs between the game board nodes. In this way, DeJong's work differs from Samuel's work[7].

DeJong admits there may be cause for concern about the ultimate size of the database. They suggest that if their initial hypothesis that " ... the number of *reasonable* board configurations *actually* encountered would be quite manageable" is incorrect then a method of forgetting could be considered[7]. For this study, it is felt that the size of the accrued database could become prohibitive and for that reason this technique will not be pursued.

The outcome of the game is the final judge of whether a player's strategy is or is not effective. This paper shows that a deterministic search to the end of a game tree is impractical. It is also impractical to attempt to evaluate every possible game board configuration prior to play and "know" which move will lead to a winning game. Because of these facts, it is important to identify some stochastic method of determining the quality of individual moves from any board configuration.

The use of different characteristics (usually unique to the game or situation) is important in the evaluation of a move. In Samuel's research, he

identified 38 parameters that can be empirically determined from any board configuration. Among these parameters are piece placement, threats, control of rows, and mobility [4].

Samuel used 16 parameters in any one evaluation function at a time. The parameters are multiplied by a coefficient computed by the program. At each move, the game board, polynomial, and the"apparent goodness"[5] of the game board are saved. Some of the coefficients are adjusted slightly at each move and the apparent goodness was checked. By backtracking through previous moves and comparing the apparent goodness the parameter can be adjusted.

If the apparent goodness was found to be not as good as was expected by a previous polynomial then the earlier game board polynomial predicted, then it must be assumed that the earlier polynomial was in error and the terms that contributed positively are given more weight and the terms contributing negatively are given less weight. Eventually if a parameter shows not to contribute significantly to a good evaluation, it was removed by swapping these terms with terms from the unused list. Using these techniques, the polynomial can be changed to hopefully improve play.

Rosenbloom also uses a polynomial to evaluate board configurations. His polynomial is based on 4 factors:

1. Edge Stability - the number of stable disks along the edge of the board.

2. Internal Stability - the number of stable disks not along the edge of the board.

3. Current Mobility - Choice of moves for that play.

[4]For a detailed list see Samuel,1981 on page 100.

[5]"Apparent goodness" was used in Samuel's research as the relative change in the positions of the two players. Although it is not clearly stated, it is assumed that the "relative change in positions" refers to the relative checker count.

4. Potential Mobility - Choice of moves for future plays.

Each parameter has a coefficient initially determined by "opinions about the relative importance of the individual components"[16]. Rosenbloom continues on to say "A limited set of variations on these values was then evaluated by playing these differing version of IAGO[6] against each other. The best variation became IAGO's evaluation function." His program is not designed in any way to demonstrate machine learning, only to examine strategies.

The first and third parameters have coefficients that are in part determined by the move number. This is done to reflect the fact that different components are more important at different stages of the game.

For this thesis, determining the stages of the game will become the sole factor in determining which strategy to use. By modifying parameters governing the evaluation of game stage, the strategy designed for each of the game stages will also be used at different times.

[6]IAGO is the name of the program created in Rosenbloom's research to play Othello.

Chapter 3

DESCRIPTION OF OTHELLO

3.1 Termini Technici

There are several terms that will be used in this thesis. Most of these terms are used in other papers. Some may be unique to this thesis.

For this paper the *board* or *game board* will refer to a regular eight by eight dimensioned playing area. It has 64 squares or *locations*. Colors of these locations are unimportant.

Othello has two *players*. A player is defined as an individual, either computer or human, who is in direct competition with the other player. Each player controls either the black or the white *disks*. A disk is an individual playing piece that is placed upon the board locations. One player placing one disk upon the board constitutes a *move*. One move by each player, for a total of two moves, constitutes a *ply*.

A distinction should be made between the *rules* of Othello and the *strategies* used to play each game. The rules are specifically agreed upon constraints for making a valid move and winning or losing an individual game. Each individual player may choose to use different secret strategies but the rules are known by both players and are explicitly stated.

The *capture* of a disc means that a disk has been changed from its existing

color to the color of the current player. This is done by *bracketing* a disk or series of disks. Bracketing is achieved when one or more opponent's discs are caught between the disc currently being played and a disc that is already on the board of the same color as the disc being played.

3.2 History and Rules of Othello

Othello, or Reversi, was first played in England in the late 1880s. Othello is played with 64 discs (one for each region on the board). Each disc is black on one side and white on the other. As the game begins each of the two players choose a color and play their discs with the appropriate color up.

The game begins with two discs for each player being placed in the center of the board as shown in Figure 3.1.

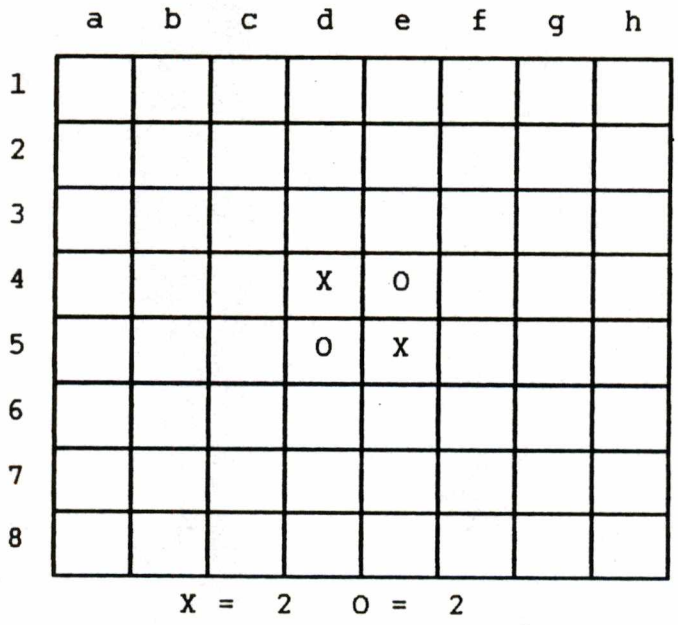


Figure 3.1: Othello Starting Configuration

For this research, the game will begin with X moving first from the position shown in Figure 3.1. X will be used to represent white and O used to

represent black.

The game proceeds with each player alternately placing a disc of his color on the board. A valid position for a new disc is a square that lies orthogonally and contiguous to discs already played. Each turn must capture at least one of the opponent's discs. Failure to meet both of these requirements results in forfeiture of that player's turn.

The goal of the game is to occupy more squares of the board than the opponent. An opponent's disc can only be captured if it is directly and contiguously bracketed between a disc of the player's color already on the board and the disc that is currently being played. These discs can be either orthogonally or diagonally contiguous. Once a disc is played on the board, a disc will always occupy that square although it may change colors during the course of the game.

Either of two conditions may be met to determine the end of the game. The first condition is that all 64 of the positions on the board have been occupied. In a well played game, this is normally the case. The second condition is that neither player is able to make a valid move where the player can capture an opponent's disc [1]. Reducing an opponent's 'mobility' or choice of moves normally causes the second condition. As long as one player can make valid moves (even if his opponent cannot), he is allowed to continue play until his opponent can finally move or he no longer has any valid moves.

Figures 3.2 - 3.7 show example board positions and valid moves with the associated captures.

As can be seen, the face of the board can change dramatically with each

[1] If a move exists, then the player *must* make a move. If a player is unable to find a valid move then based on prior agreement of both players, the player who is in violation of the rules may have to suffer any agreed upon penalty for violation of the rules including forfeiture of the game. This is an example of Von Neumann's theory of *rational behavior*[21]. It would also make sense that assuming rational play among both players that each player would take every opportunity to better his standing in the game.

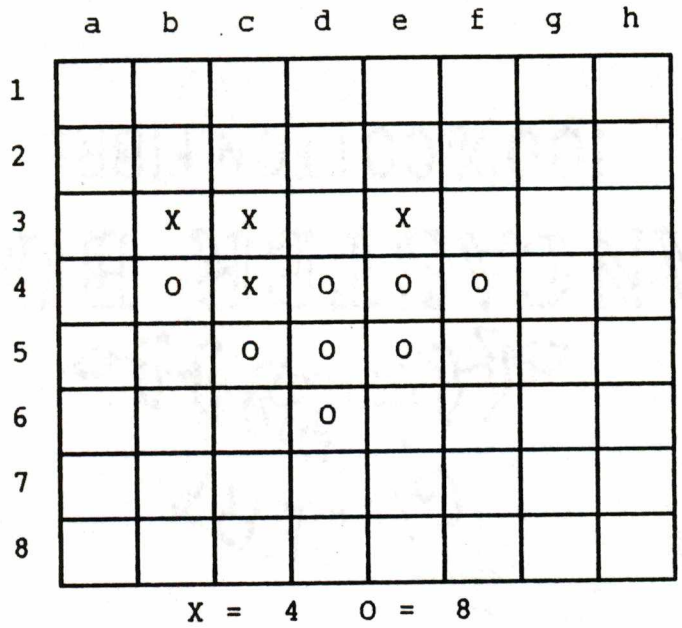


Figure 3.2: Example 1–Before

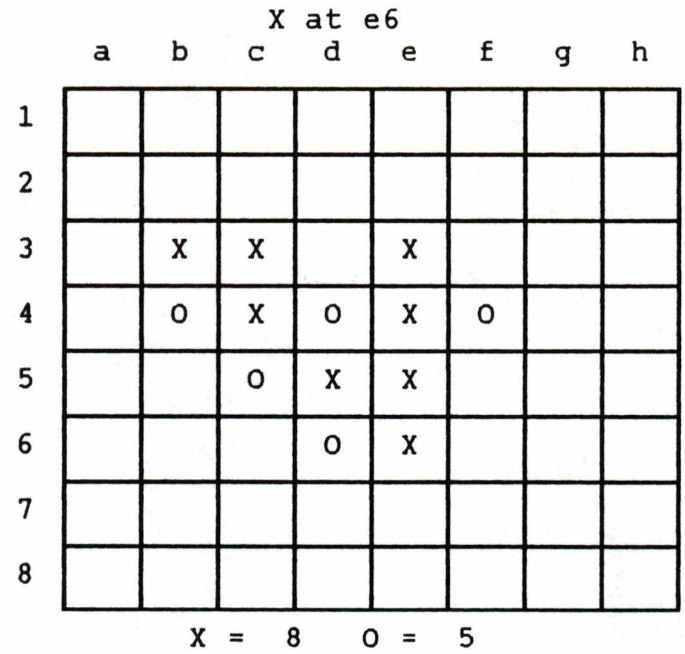


Figure 3.3: Example 2–After

With the board configured as in Figure 3.2, *X moves to e6. This brackets the O discs vertically between e6 and e3 capturing e5 and e4. The O disc at position d5 is also bracketed on the major diagonal between the newly played e6 and the disc at c4. This leaves the game board configured as in* Figure 3.3.

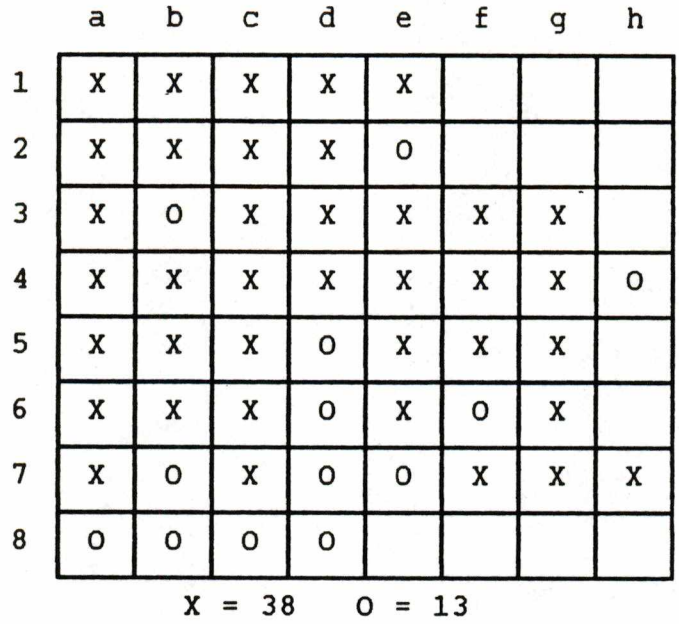


Figure 3.4: Example 2–Before

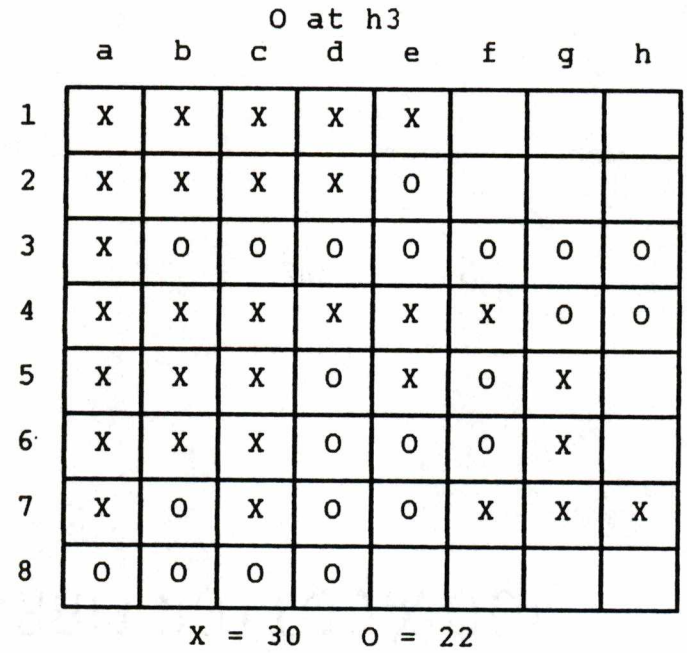


Figure 3.5: Example 2–After

At a later stage in the game, the board is configured as is shown in Figure 3.4. *A play by O at h3 brackets X discs horizontally between h3 and b3 and along the minor diagonal between h3 and d7. This captures c3,d3,e3,f3 and g3 along with e6,f5 and g4. With this single move, O improved its disc count from a 25 disc deficit to a mere 8 disc deficit. This is shown in* Figure 3.5.

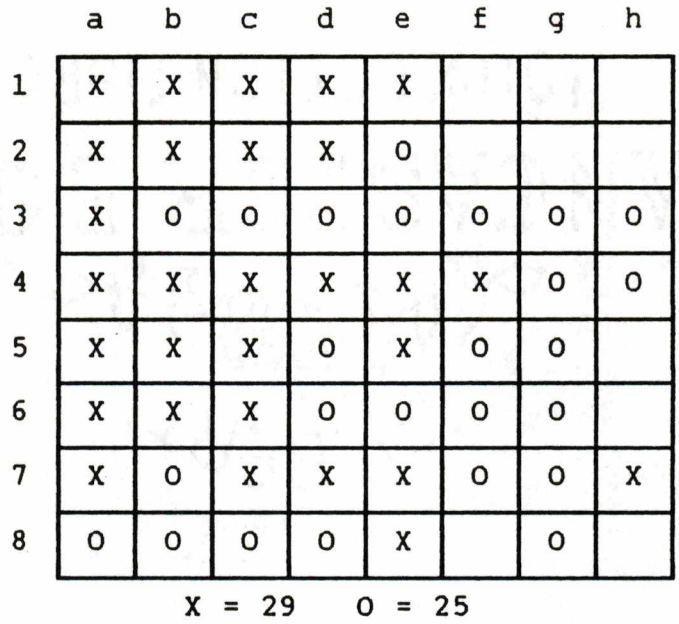


Figure 3.6: Example 3–Before

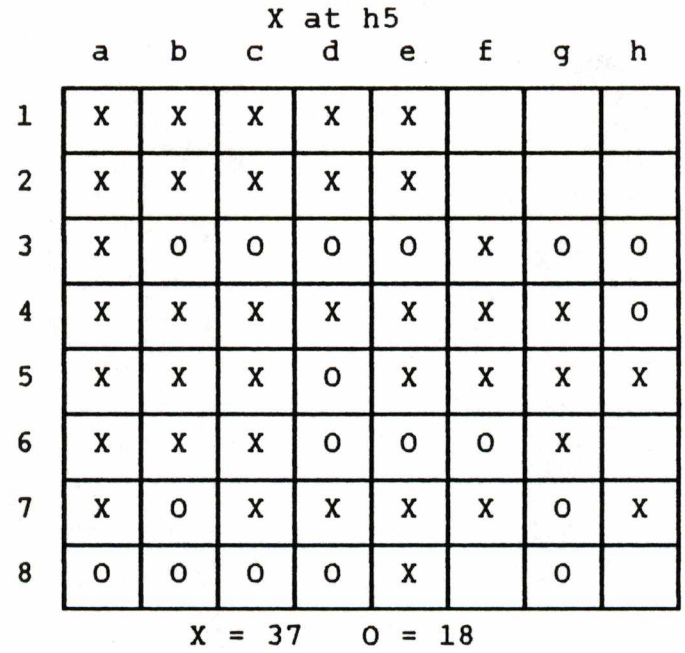


Figure 3.7: Example 3–After

Figure 3.6 *and* Figure 3.7 *show a dramatic play and change in disc count along 3 lines of capture. With the board configured as in* Figure 3.6, *X plays at h5 and brackets e2,f3 and g4 along the major diagonal, f5 and g5 along the horizontal and f7 and g6 along the minor diagonal. This results in a change from a 4 disc advantage to a commanding 19 disc advantage in X's favor.*

disc played. As many as 19 positions can change colors in a single play. Normally, the number of positions changing colors is about half that number.

There are characteristics that make Othello unique among games being used in game theory research. The game of Othello, although it uses the traditional checkers board, is unorthodox in the fact that once a position is occupied, that position will never become unoccupied. Each position can be played, captured, and then recaptured by the opposing players several times during the course of the game.

This characteristic of an ever increasing number of involved game pieces makes Othello unique. A "stable" disc is one that cannot change its colors. This happens when a disc is positioned such that it can not be recaptured. Until many discs become stable, the disc differential can change dramatically during the course of the game and even with individual moves. Although a maximum disc count is the goal of the game, using a maximum disc count as the sole heuristic for determining the quality of an intermediate board may not always be reliable[16].

> As the end of the game nears, the primary concern is with maximizing the disc differential. However, as was shown by the failure of the maximum disc strategy, flipping a disc does little good if the opponent can immediately flip it back. The concern must be with maximizing the *final* disc count.

Another characteristic that is advantageous for the machine learning researcher working with Othello is that every disc being played is subject to the same rules for valid moves as any other disc. In chess, there are 7 distinctly different types of valid moves, depending upon the piece being moved. Even in checkers, a "king" has different valid moves than the regular pieces. This makes programming and analyzing the rules of Othello moves somewhat less complex and thus the effort can be concentrated on the machine learning aspect of the game. It is for this very reason that Samuel chose checkers over chess[17] in his

research and this researcher feels that Othello is a better choice over checkers for this thesis.

Chapter 4

USING WEIGHTED SQUARES IN OTHELLO

4.1 Strategies to Play Othello

Although Othello is a relatively recent entry into the field of machine game playing, some strategies seem to be emerging attempting to play the game more effectively. In his research on the game of Othello, Paul Rosenbloom identified three main strategies[16]:

1. A maximum disc strategy where the sole impetus behind choosing a specific move is to capture the greatest number of discs.

2. A weighted square strategy is the realization that not all squares are of equal importance (due to location). Different squares may be worth proportionally more.

3. A minimum disc strategy where a player attempts to minimize his number of discs at least initially in order to increase his mobility later in the game.

Maximum disc strategy, because Othello is nondeterministic in practice, is impractical because it is too greedy in the short term. It completely ignores any type of planning beyond its search range in determining the placement of

the new discs. It is important to note, however, that the objective of the game is for a player to maximize his final disc count and minimize his opponent's discs. Rosenbloom and this researcher agree on the observation that the maximum disc strategy should become the dominant strategy when approaching end game situations. It should also be obvious that the only way that this can realistically be approached is to have discs in place that optimize this goal before the end game strategy takes over. This will be discussed more fully in Chapter 4 when the different game stages (opening game, middle game and end game) are distinguished.

When research for this project began, strategy for the program in this research was based solely on a maximum disc strategy. This strategy provided an acceptable level of play and was able to beat most players who were novices at the game. After the human had played approximately five to ten games, the program's weak strategy became obvious. Even a novice player realized the fact that some squares provided an anchor from which more disks could be captured. The program appeared to dominate in the intermediate stages of play, but when the final plays were made, the human opponent was able to recapture most of the discs and win the game. Even with a decrease in the time required to evaluate each move and an increase in the depth of the mini-max search, the program played poorly against humans with even a little experience.

The strategy of weighted squares overcomes the problem of assuming all the squares on the board are of equal importance. A maximum disc strategy is a type of weighted square strategy where each square has an equal value. Emphasis of some squares on the board can be achieved by giving them a value greater than other squares. By totaling the values of the squares instead of the number of squares, a strategy is developed to occupy more valuable squares and avoid less desirable squares.

The first goal of this research is to show that an uninformed weighted square strategy can learn to compete with a more sophisticated informed weighted square strategy. To do this, the program playing the uninformed strategy modifies its square values after playing against the more complex strategy. If it can be shown to improve its performance against the more sophisticated strategy, then it will have demonstrated machine learning.

The weighted square strategy is used to distinguish more desirable squares from less desirable squares. These weights are based upon expert opinions of which squares are more important to control. Since the weighted square strategy is an efficient method of distinguishing square values, it is the strategy that was initially investigated in this thesis.

Two weighted square maps were identified. Figures 4.1 and 4.2 show square weightings suggested by the Invicta rule book[9] and by Paul Arnold[1]. In an effort to determine the better of these two, the strategies were played against each other. A mini-max tree with $\alpha - \beta$ pruning is used for this as well as all other search strategies throughout this research. The search is taken to a constant depth of 3 ply [1] which generally conforms to the suggested 90 minute playing time.

Figure 4.3 shows the disc differential[2] as the game progresses. As the results of this game shows, the Invicta strategy is a better weighting strategy when played against the Arnold weighting strategy. For the first third of the game, neither strategy takes a decisive lead. In the next third of the game, the Arnold's strategy takes somewhat of a lead. Although the advantage is slight, it does not relinquish this lead until the 54th move when Invicta takes the lead

[1] 3 moves by each player.

[2] A common practice in reporting games of Othello is to show the difference in the number of discs controlled by each player after each turn in the game. This is done by subtracting one player's total discs from the other players. In this example the number of discs controlled by the Arnold's strategy is subtracted from the number of discs controlled by the Invicta strategy.

	a	b	c	d	e	f	g	h
1	15	10	10	10	10	10	10	15
2	10	2	2	2	2	2	2	10
3	10	2	5	5	5	5	2	10
4	10	2	5	5	5	5	2	10
5	10	2	5	5	5	5	2	10
6	10	2	5	5	5	5	2	10
7	10	2	2	2	2	2	2	10
8	15	10	10	10	10	10	10	15

Figure 4.1: Invicta's weighted square strategy

	a	b	c	d	e	f	g	h
1	4	2	4	3	3	4	2	4
2	2	1	3	3	3	3	1	2
3	4	3	4	4	4	4	3	4
4	3	3	4	4	4	4	3	3
5	3	3	4	4	4	4	3	3
6	4	3	4	4	4	4	3	4
7	2	1	3	3	3	3	1	2
8	4	2	4	3	3	4	2	4

Figure 4.2: Arnold's weighted square strategy

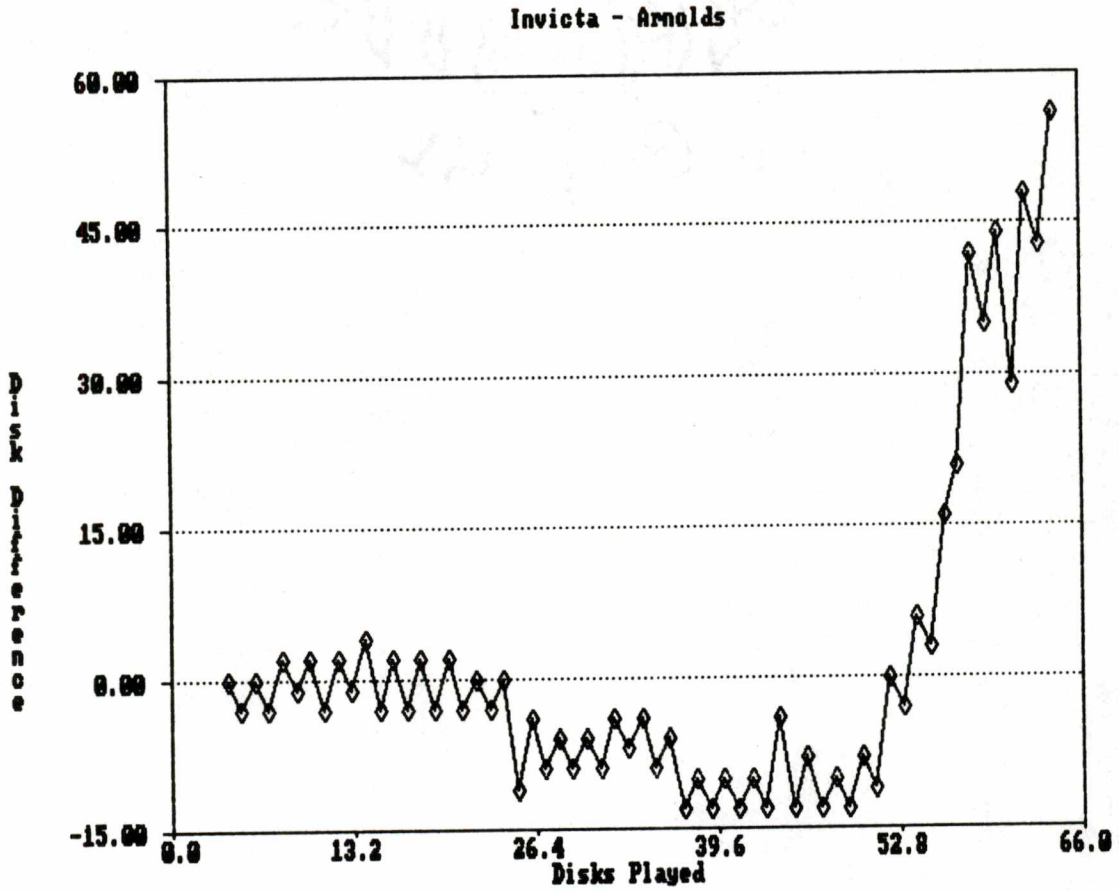


Figure 4.3: Disc differentials

Results of this competition resulted in a victory for Invicta's strategy by a score of 60 discs to 4. The disc differential after each turn is shown in Figure 4.3.The values shown are the current number of Invicta's strategy discs minus the number of Arnold's strategy disks.

and wins by a large margin. Reversing the strategies (allowing Invicta to play first) resulted in similar results.

The two strategies were also played against one another at various other depths with inconclusive results. Neither strategy won by any significant (more than 5 discs) margin at the other depths tested. As a result of the difference at 3 ply, the Invicta strategy was chosen as the informed strategy from which an uninformed weighting strategy would be taught.

An important advantage of the weighted square strategy is that the amount only 128 bytes of data needs to be kept. The time to evaluate this data is proportional to the amount of time used by the computer to examine each element in the array and add its value to a total. This gives the advantage of short evaluations and thus more evaluations can be made at each turn.

4.2 Methods of Improving the Play

Now that an informed strategy is identified (a weighted square strategy), a method of teaching an uninformed strategy must be determined to improve the performance of the Othello playing program. Paul Hoffman identifies 3 methods of learning related to this strategy in his work with the game of MILL[10].

The first method Hoffman suggests is "from observation". A program watches the moves of its human or machine opponent. If its opponent is experienced, the mill playing program (MEL) will get good examples and thus play more skillfully. If MEL's opponent is inexperienced, MEL will learn bad examples and thus play more poorly.

The second learning technique is "events from a teacher". In this method, events are codified directly into the learning program's database. This method, although it shows better play when compared to a program without the codified experience, is not consistent with the concept of machine learning. It is not

gained knowledge but rather ingrained knowledge.

The third technique is the one which was employed for this phase of the research into Othello. It is the knowledge "gained from experimentation". MEL records each move that is made during the course of any game. If it turns out to be a favorable position [3], then the moves are saved as good moves.

GEM is an induction program developed by Intelligent Systems Group at the University of Illinois at Urbana-Champaign that is used in conjunction with MEL. GEM is capable of deducing rules from observations based on the moves that have been recorded. An explanation of this system can be found in [13].

As is reported in previous papers concerning Othello, development of a large group of rules defining favorable moves does show improved play[7] but this thesis will not pursue a rule deduction strategy for machine learning. This stage of the research is only designed to show that play can be improved with very simple machine learning techniques.

4.3 Modifying the Weighted Squares

Since the Othello board can change so quickly, it is hard to identify individual moves that lead to a winning game. Favorable positions during the course of the game are thus hard to identify. For this reason, a won game was used as the criteria for identifying desirable squares. The positions occupied at the end of the game are the positions that are labeled as desirable.

This research uses an informed or "expert" weighting strategy and an uninformed "student" weighting strategy. The informed strategy has weights assigned to each of the squares as shown in Figure 4.1. The student has each of

[3] "Definitions of favorable positions must be provided, but they can be as simple as a won game"[10].

the squares weighted equally with a value of one, hence it played a maximum disc strategy initially. The goal of this phase of the research was to make the uninformed search strategy imitate and eventually become competitive with the informed weighting strategy.

Results of this form of learning can be seen immediately. From the first game, the learning system begins modifying its square weights to alter its choice of moves for subsequent games.

The expert and the student begin playing games against each other at a search depth of three moves. After each game is completed, the board is analyzed and, regardless of which player wins, the student's weighting of each of the squares controlled by the winner is increased by one. Once this is done, the square values are reduced using Equation 4.1:

$$S_{c,r} = S_{c,r} - (l - 1) \tag{4.1}$$

where S is the set of weightings of the squares, c and r are the column and row of the individual location and l is the lowest square weighting on the board.

Learning techniques that cause the uninformed weighting program to learn quickly unfortunately have the side effect where the uninformed weighting strategy can easily be misled by squares that may not be good squares to control but happen to have been controlled by the winner. This is an example of using a very low *threshold of acceptance*. One game being won with a particular location being controlled is all it takes to increase the value of that location. Hoffman states a similar situation concerning MEL[10]:

> If the threshold were two, for example, no event would be recorded until it had twice led to a favorable position [*in the game of MEL*]. (The idea of thresholds is due to R.S. Michalski.) A high value for the threshold provides high quality events at a slow pace. A low value provides many events of lesser quality.

The situation of "lesser quality events" is evidenced in the first exper-

iments with Othello. The first series of games between the informed strategy and the uninformed learning strategy showed the game disc differences [4] shown in Figure 4.4.

The final weightings for the student are shown in Figure 4.5. It should be noticed that the student places a greater emphasis on the locations along the edge just as the informed strategy does. It should also be noticed that the weightings are nonsymmetrical and several locations on which the informed strategy does not place a lot of importance are emphasized in the student's final weightings. Statistical analysis of these weightings compared with the weightings in Figure 4.1 shows an average difference of 2.93 and a standard deviation of 2.384.

Heavier weightings in the upper left corner of the student's final strategy reflects the order in which the available moves are evaluated by the expert. It was hoped that the uninformed search strategy could take into account the order in which the informed strategy searched for its best move and be able to improve play over the informed strategy in respect to this search sequence. This shows not to be a significant factor.

The learning program was then modified to separate the board into four symmetric quadrants bisecting the board horizontally and vertically. Every position was dependent upon its three symmetrically equivalent positions. Figure 4.6 shows the location of the equivalent squares. With this strategy, a threshold greater than one could be implemented. This provides *a high threshold of acceptance* which in this case is a threshold of 3. If three or more equivalent positions are occupied by the winner of the game then the value of all of the equivalent positions are increased by one.

[4]These are the differences at the end of the game. In these examples the number of discs controlled by the informed strategy is subtracted from the number of discs controlled by the uninformed strategy.

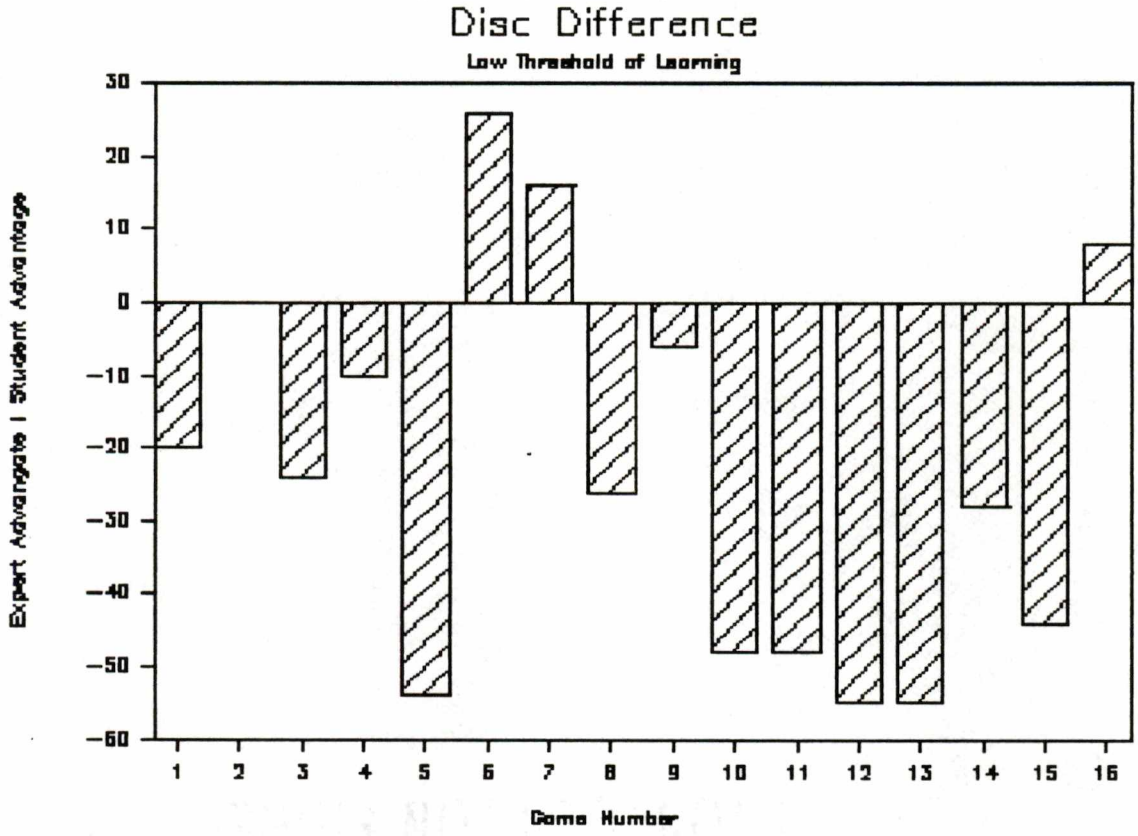


Figure 4.4: Disc difference with low threshold of acceptance.

	a	b	c	d	e	f	g	h
1	12	13	13	11	13	12	10	10
2	11	10	11	8	6	8	8	10
3	11	6	10	9	7	8	8	10
4	11	10	7	7	5	6	8	10
5	11	7	10	6	11	6	7	10
6	11	6	7	7	6	12	1	9
7	11	8	7	4	5	2	5	6
8	11	10	10	10	10	6	7	9

Figure 4.5: Final student weightings with low threshold of acceptance.

	a	b	c	d	e	f	g	h
1	A	B	C	D	D	C	B	A
2	E	F	G	H	H	G	F	E
3	I	J	K	L	L	K	J	I
4	M	N	O	P	P	O	N	M
5	M	N	O	P	P	O	N	M
6	I	J	K	L	L	K	J	I
7	E	F	G	H	H	G	F	E
8	A	B	C	D	D	C	B	A

Figure 4.6: Equivalent disks.

Hoffman employs a similar technique by normalizing each move in a game of MILL into one of the 8 symmetric parts of the MILL board. This way, due to the symmetry of the MILL board, equivalent *moves* can be identified.

For example, at the end of a game the winner of the game occupied c2, f2, c7, a2 and h6 among other locations. Since c2, f2, and c7 are equivalent positions, c2, f2, c7 and f7 will have their square values increased. On the other hand, a2 and h6 will not have their square values increased because only two of the equivalent squares are occupied.

A problem with this strategy arises after about 10 games. At this point, the informed strategy beat the uninformed strategy by 54 disks. In this situation, all sets of positions are above the threshold and were thus increased in value. Once the board values are normalized, the board values are unchanged from their value as the game began. This causes the subsequent games to play exactly as the previous game had played and the uninformed strategy stopped learning.

To alleviate this problem, the idea of "mutations" is introduced to the learning strategy. This idea came from a talk given at the University of Tennessee by Gunar Liepins, a researcher at Oak Ridge National Laboratories. Dr. Liepins' talk concerned a paper that is to be published on genetic algorithms. In his paper he suggested that periodically, to produce variation in the population, random bits of the population bitstrings be changed[12].

Although this will not work directly, a variation of this technique can be applied to this research. For the Othello program, in the event that less than 3 or more than 14 sets of equivalent positions are changed, then the values of 4 random sets of positions are increased by one. This produces the change needed to prevent the weightings from stagnating.

For example, referring to Figure 4.6, if the expert should win by a large enough margin, such as taking 60 of the 64 discs, then the expert could control

at least 15 of the 16 sets of equivalent discs. Once these locations are increased by one then the overall change to the board is the same for all locations except for 4 of the squares. To make a larger change 4 of the sets of equivalent squares (A through P) are chosen at random. These 4 sets of squares (16 squares in all) each have their values increased by one. This means that 20 squares would have their values changed and would offer a chance for variation in the weightings of the uninformed weighting strategy.

Using the concept of mutations produces the game differences shown in Figure 4.7.

Toward the end of the run of games, it should be noticed that the student consistently wins by an increasingly larger margin.

After a series of wins, the student loses to the informed strategy. This is explained by the mutations. The program is designed so that the mutation algorithm doesn't distinguish between the informed strategy winning or the uninformed strategy winning. If mutations did not affect the student, then should the student win by a very large margin, no weightings would be changed and all future games would play exactly as the previous game did.

Examining the final square weightings for the student in Figure 4.8 shows the importance that the uninformed strategy eventually placed on each of the locations. Statistical analysis of these weightings compared with the weightings in Figure 4.1 shows an average difference of 2.80 and more importantly a standard deviation of only 1.589. This is a significant improvment over the earlier experiments.

4.4 Weighted Square Strategy Conclusions

The experiments using a low threshold of acceptance (a threshold of one) prove to be only a partial success. It is functional in that the quality of the play of

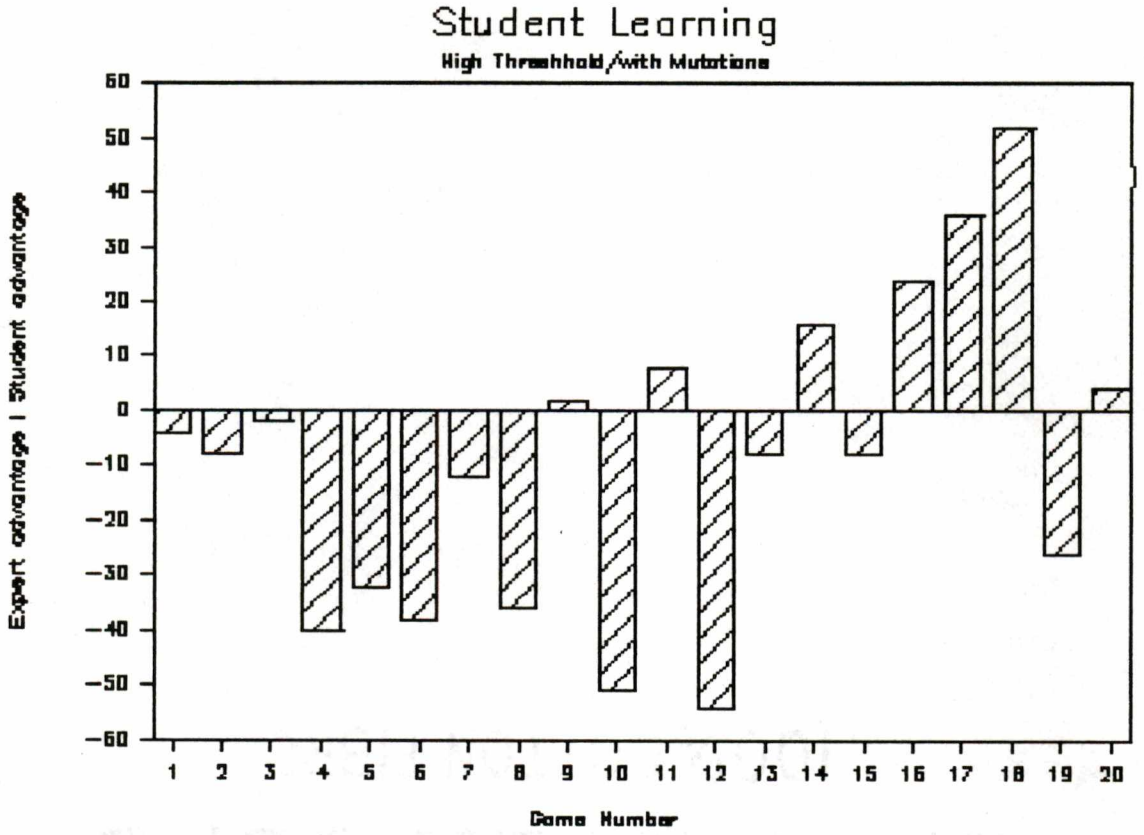


Figure 4.7: Disc difference with high threshold of acceptance and mutation.

	a	b	c	d	e	f	g	h
1	9	8	5	5	5	5	8	9
2	9	5	4	4	4	4	5	9
3	9	4	8	1	1	8	4	9
4	8	2	9	8	8	9	2	8
5	8	2	9	8	8	9	2	8
6	9	4	8	1	1	8	4	9
7	9	5	4	4	4	4	5	9
8	9	8	5	5	5	5	8	9

Figure 4.8: Final weightings with high threshold of acceptance.

the uninformed strategy is changed. However, this low threshold of acceptance does not improve the play.

A substantial improvement in the performance of the strategy is realized when the threshold of acceptance is increased to 3. The improvement is apparent when the student's weightings are compared to the origninal Invicta strategy. Using a low threshold of acceptance, the standard deviation between the weightings is a relatively large: 2.364. With a high threshold of acceptance, the standard deviation is a much lower: 1.589.

Increasing the threshold of acceptance causes the uninformed strategy to slowly but steadily improve its performance against the informed strategy. Using this technique without mutations, though, leads to the uninformed strategy reaching an artificial plateau where it stops learning. Mutations cause random changes in the uninformed strategy values. These mutations may not always be beneficial but the changes cause the program's trying new weightings in an

attempt to improve the learning program's performance.

This series of experiments shows that an uninformed strategy can improve its performance over a series of games. It also shows that that the *threshold of acceptance* must be sufficiently high to prevent the program assigning credit to squares that may not lead to a winning strategy. "Mutations" also prove to be a valid technique to introduce variation into a population of values that may otherwise remain unchanged.

More importantly, these experiments provide two techniques that are used in the next series of experiments. The first technique is the idea that an uninformed computer program can improve its performance by competing against an informed computer program. This eliminates the need for a human opponent that can accidentally mislead the uninformed strategy.

The other technique that is employed in the next experiments is the use of the final outcome of the game as the criteria for modifying the uninformed strategy's values. This provides a basis for determining which player's strategy is stronger and thus provides input on how the uninformed strategy should be modified.

Chapter 5

CLASSIFICATION OF GAME STAGES

Each time a game is played, the play of that game [1] progresses in a normal fashion from the opening game to the middle game and finally to the end game stages. The determination of what stage of play the game board configuration is in at any one instance during an individual game is the subject of this part of the research.

5.1 Definition of the Fuzzy Functions

Some functions are important tools in this thesis. The three functions [Equations 5.2,5.3, and 5.4] that are described will help to quantify characteristics of Othello that are are most important during the opening game, the middle game and the end game.

Examples would best demonstrate the use of fuzzy sets and their representation in terms of membership functions. Suppose it is known by empirical testing that a room is definitely crowded when 75 people are in the room and the room is definitely not crowded when 25 people are in the room. To determine whether or not a room is "crowded", it would be logical to say that it *definitely*

[1] "play" in this instance referring to all of the individual moves that take the board from the opening configuration to a condition constituting the end of play.

is crowded or it *definitely is not* crowded. When there are fewer than 25 people or more than 75 people, the definition of whether or not the room is crowded is easy. What about if the room has 50 occupants? How about 30 occupants? How about if it has 70 occupants? How do you define the room in each of these conditions?

As was mentioned in Chapter 3 of this thesis, it is not always correct to define *hard* sets. Fuzzy sets are much more useful in situations where a simple *yes* or *no* is insufficient in describing the membership of an element.

Obviously, to say that the room is crowded when there are 26 people in the room, or to say that the room is not crowded when there are 74 people in the room is inaccurate. For this reason the room at any stage of occupancy must be given a *degree of membership* in the set referred to as a "crowded" room.

Bezdek proposes a stair step type function (5.1) to compute the degree of membership that an element shows in a fuzzy set[3].

$$\forall u \in X; M(u) = \begin{cases} 1, & 75 \leq u \\ 0.95, & Q_1 \leq u < 75 \\ 0.90, & Q_2 \leq u < Q_1 \\ \vdots & \vdots \\ 0.05, & 25 \leq u < Q_n \\ 0, & u \leq 25 \end{cases} \tag{5.1}$$

where u is the occupancy of the room, $M(u)$ is the degree of membership in the set "crowded" room and Q_1 through Q_n are values chosen between the minimum and maximum values for occupancy, in this case 25 and 75.

To make the classification smooth and differentiable, a different function becomes useful. Zemankova uses as a basis for her work a series of equations that are useful to this thesis. Based on Zadeh's work[22,23,24], Zemankova states the following[25]:

...it is convenient to be able to express the membership function M_a as a standard function whose parameters may be adjusted to reflect the subjectivity of the characteristic function. A standard function of this type that maps a fuzzy subset of a real line onto the interval [0,1] is the S-function, which is a piecewise quadratic function defined as follows:

$$M(u) \triangleq S(u;a,b,c) \begin{cases} = 0 & for\ u \leq a \\ = 2\left(\frac{u-a}{c-a}\right)^2 & for\ a \leq u \leq b \\ = 1-2\left(\frac{u-c}{c-a}\right)^2 & for\ b \leq u \leq c \\ = 1 & for\ u \geq c \end{cases} \tag{5.2}$$

where the parameters a and c are the lower and upper fuzzy subset interval bounds, respectively, and the parameter $b = \frac{(a+c)}{2}$ is the crossover point, i.e. the value of u at which $S(u;a,b,c) = 0.5$."

The function in Equation 5.2 creates a degree of membership $M(x)$ where $0 \leq M(u) \leq 1$. A membership value of 1 indicates that the room *definitely is* in the set labeled as "crowded". A membership value of 0 indicates that the occupancy is *definitely not* in the labeled "crowded". Values between 0 and 1 indicate partial membership in the set of crowded. Figure 5.1 graphically shows this membership for occupancies from 0 to 125.

Similarly the set of a "not crowded" room can also be determined. By taking the negation of the equations (Equation 5.3), a descending function is created as shown in Figure 5.2. For this example, a value of 1 indicates full membership in the set "not crowded" and a membership of 0 indicates that the room is considered to be not in the set "not crowded" or in reality the room is *crowded.*

$$M(u) \triangleq S(u;a,b,c) \begin{cases} = 1 & for\ u \leq a \\ = 1-2\left(\frac{u-a}{c-a}\right)^2 & for\ a \leq u \leq b \\ = 2\left(\frac{u-c}{c-a}\right)^2 & for\ b \leq u \leq c \\ = 0 & for\ u \geq c \end{cases} \tag{5.3}$$

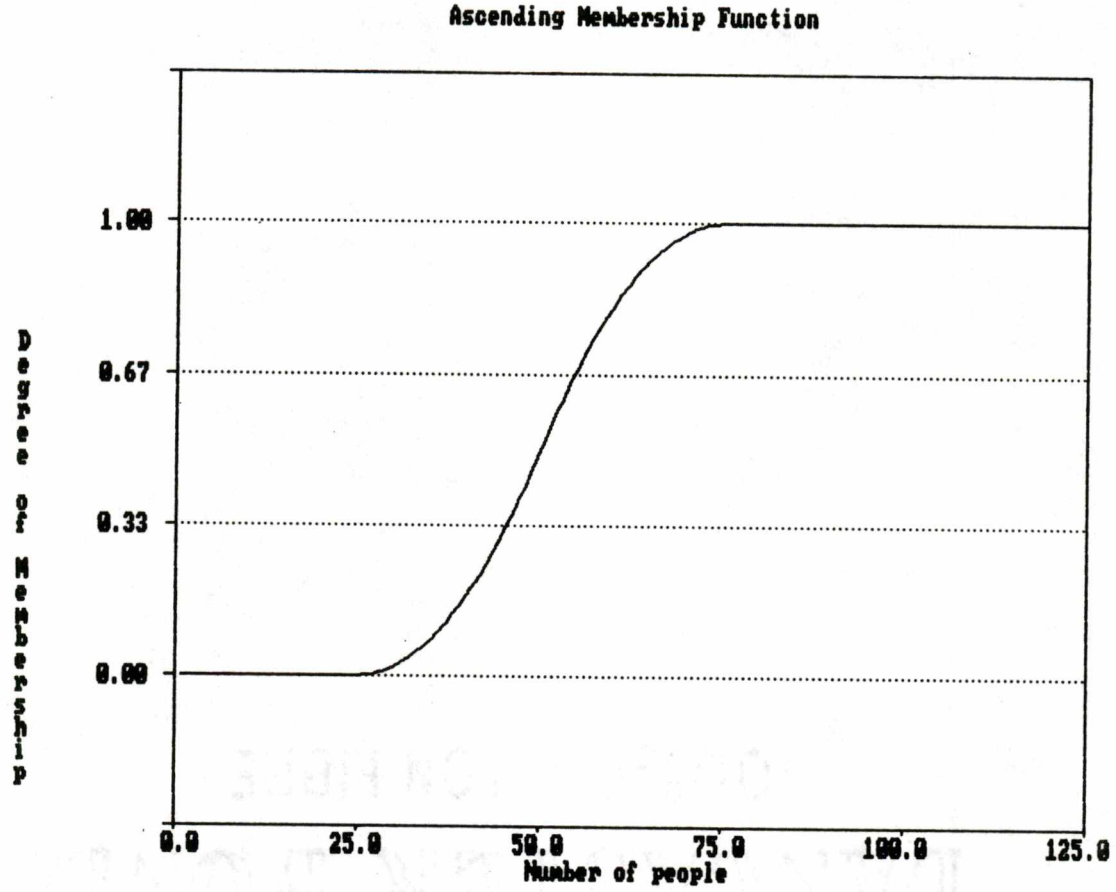


Figure 5.1: Ascending S-graph for the set *crowded.*

Equation 5.2 is a membership function that produces the graph shown in Figure 5.1 for the set of crowded.

A third membership function is necessary to define a fuzzy set with a local maximum. An example of this would be to determine whether or not a room is "comfortably occupied". Assuming that no one likes to sit in a large empty room or in a room that is overly crowded, there must be some occupancy that is optimum for comfort. This $M(x)$ is easily defined as a *bell* function. Equation 5.4 shows the function used to create the graph in Figure 5.3.

$$M(u) \triangleq S(u;t,b) = e^{-\left(\frac{u-t}{b}\right)^2} \tag{5.4}$$

This function requires different parameters. In this parametric equation (Equation 5.4), t is the point at which $M(x)$ reaches its maximum and b is the "degree of fuzziness". For this example t is 50 and b is 4.

5.2 Fuzzy Components for Othello

Determination of the game stage is critical because different strategies are used at each of the different stages of a game. To determine the stage of a game, more must be considered than simply the number of moves that have taken place or the amount of time required for the search as Rosenbloom suggests. To determine in chess, for example, the stage of a game by simply counting the number of moves that have taken place can in fact be erroneous. Although it is unlikely, it is possible for a player to move a piece from its original position on one move then move it back on his next turn. If a player continued to do this, after any number of moves evenly divisible by 2 the player's pieces would be in their original configuration. This clearly does not show progression through the game.

In this example, if the first player played a game as described above, and

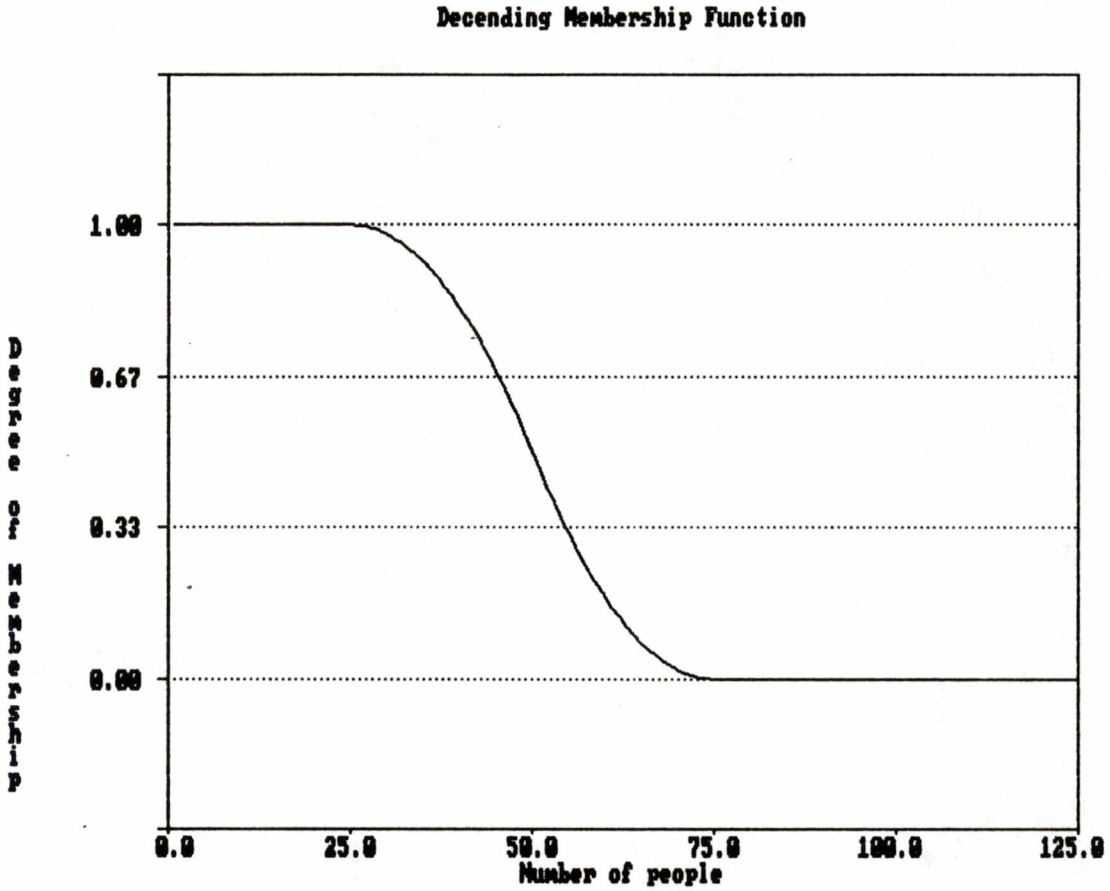


Figure 5.2: Descending S-Graph for the set *not crowded.*

Equation 5.3 is a membership function that produces the graph shown in Figure 5.2 for the set not crowded.

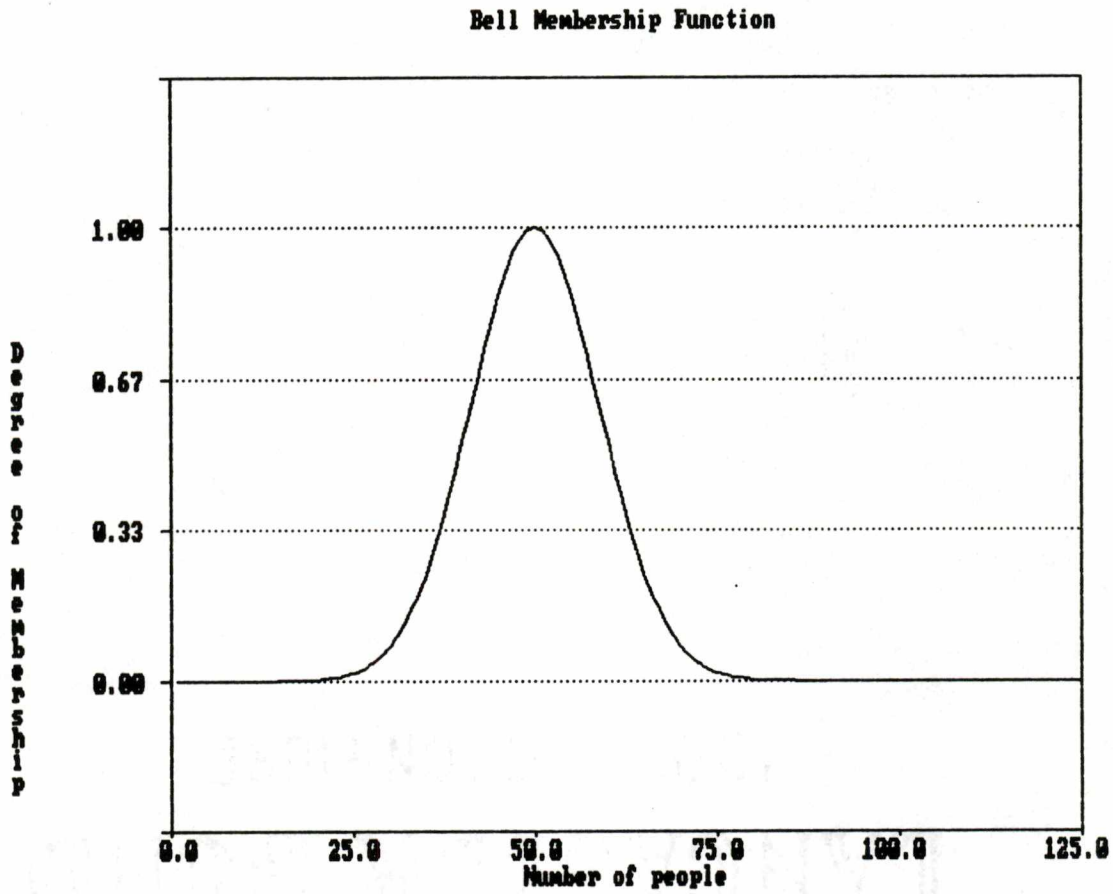


Figure 5.3: Bell S-Graph for the set *comfortably crowded.*

Equation 5.4 is a membership function that produces the graph shown in Figure 5.3 for the set comfortably crowded.

the second player arbitrarily started a middle game strategy after 20 moves and an end game strategy after 50 moves, then the second player's end game strategy would be facing an opponent whose pieces were in the original configuration. This may cause the second player's strategies to falter.

When classifying a game stage, it is important to notice other factors besides the number of moves (although the number of moves may be a factor). Bezdek says about classification[3]:

> The purpose of classification may not be as simple as construction of a taxonomy of the physical processes involved; or as complex as deciding a course of action based on identification of a distinguished individual. Classification attempts to discover associations between subclasses of a population.

For this research, the game board is divided up into three regions. The innermost locations are called the "mid-locations" (labeled m). The locations along the edges of the board are called "edge-locations" (labeled e). Between these two areas is the "frontier-locations" (labeled f). Figure 5.4 shows how these locations are distributed.

It should be noticed from Figure 5.4 that there are 16 mid-locations, 20 frontier-locations and 28 edge-locations.

Any factors associated with a game may be used to determine the game stage. For this research, a physical characteristic of the game of Othello is used in classifying the game stage. Referring back to Figure 3.1 will show the starting configuration for each game. A rule of Othello states that pieces must be placed contiguous to pieces already played. As the game progresses, the group of discs played will grow towards the edges of the game board.

This means that the pieces that are played at the beginning should center around the 4 discs that are initially on the board. These moves could be placed in a set called "start moves". Locations that should be occupied toward the middle of the game and are typically further out from the center can be classified as

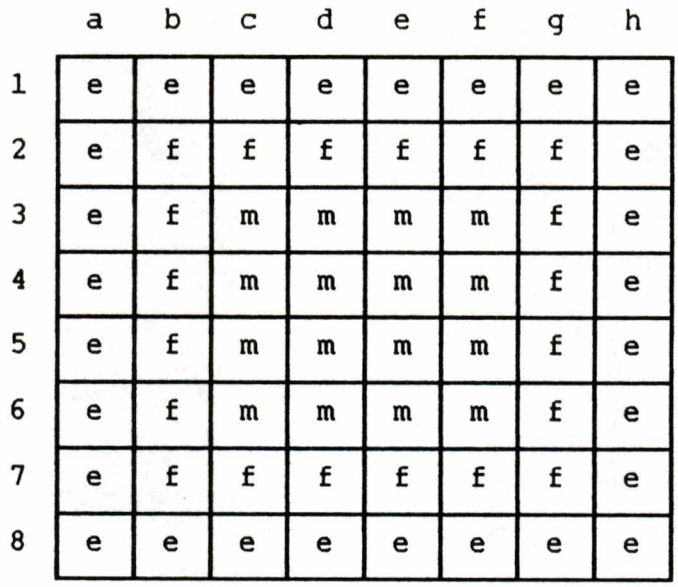


Figure 5.4: Regions for stage classification

"middle moves". Locations that should not be occupied until the final stages of the game can be called "late moves". These sets form the basis of the game stage determination.

In a typical opening game situation, as the number of discs in the mid-locations increases the degree of membership in the set "start moves" will decrease. As discs begin to be placed in the frontier-locations of the board, the degree of membership in the set "middle moves" also increases. As the frontier-locations begin to fill up, the degree of membership in the set "middle moves" starts to decrease. Discs placed in the edge-locations indicate that the game is approaching the "late moves" situation. As the number of discs in these location increases, the degree of membership in the set "late moves" will also increase.

Referring back to Equation 5.2, Equation 5.3, and Equation 5.4, it will be noticed that the lower (a) and upper (c) bounds are required to define the graph. These bounds were set to the minimum and maximum number of discs

that are possible in each of the middle, frontier and edge locations. For the middle locations, the lower bound was set to four and the upper bound was set to 16. Remember that the game starts with four discs in the center of the board (all within the start-moves region). For the frontier locations, the lower bound was set to zero and the upper bound set to 20. For the edge locations, the lower bound was set to 0 and the upper bound set to 28.

For Equation 5.4, a factor to determine the degree of fuzziness was needed. Four was arbitrarily chosen because it gave an acceptable curve with which to work. It should be noted that the only reason for the choice of this value is the fact that it yields satisfactory results for this series of experiments.

As discs are placed onto the board in a typical game, the degree of membership of each of the sets *start moves, middle moves* and *late moves* was computed and plotted on Figure 5.5.

A *typical game* for the purposes of this thesis, is a game where identical strategies are competed. In the case of Figure 5.5, a game using the weighted square strategy defined in the first stage of this research (competing the Invicta weightings against itself) is replayed . At each move the board is analyzed to determine the degree of membership in the sets of start-moves, mid-moves and late-moves. These values are recorded and are shown in Figure 5.5.

5.3 Combination of Components

It is important to determine which stage of the game the current board configuration is in so that the proper strategy can be applied. To do this the degree of membership of each of the game stages (opening, middle or end) is computed as a combination of each of the individual sets start-moves, mid-moves, and late-moves. The set of start-moves is a function of discs placed in the mid-locations. The set of mid-moves are a function of the number of discs placed in the frontier-

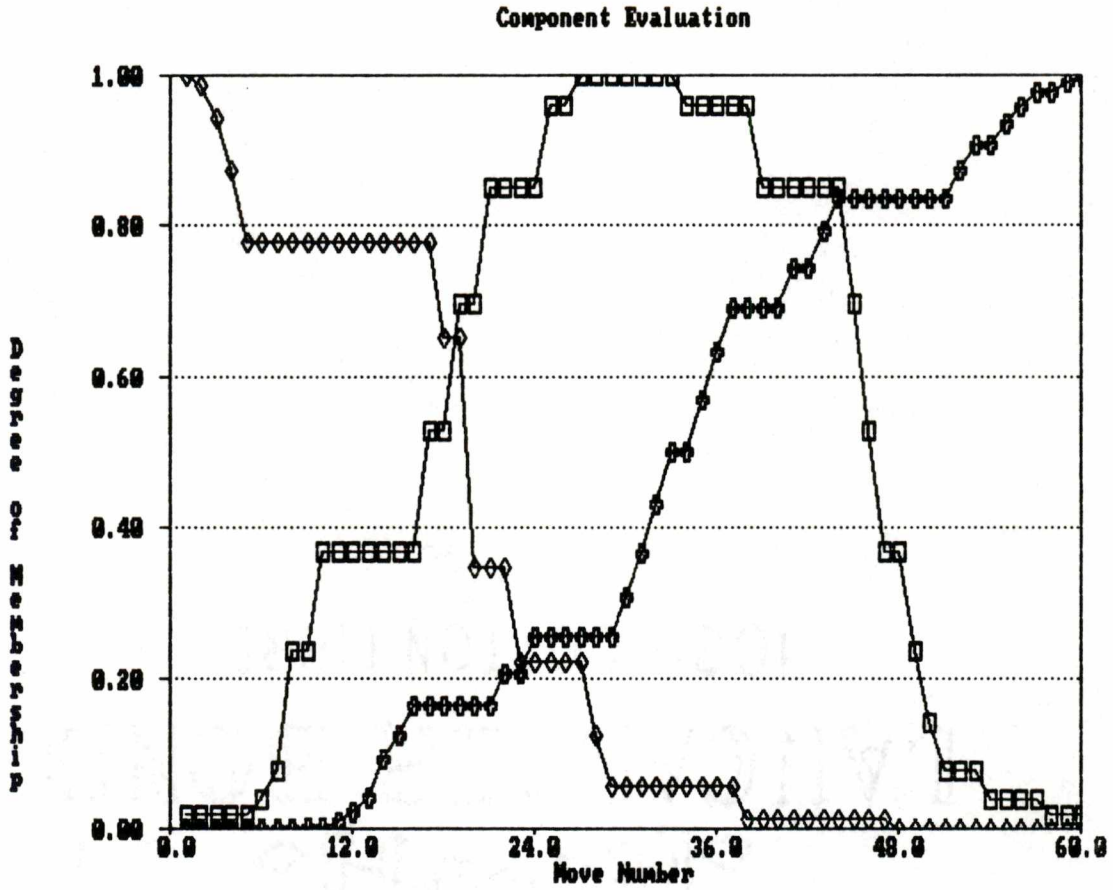


Figure 5.5: Degree of memberships in Move Sets

Figure 5.5 shows the degrees of membership that each of the regions demonstrate in the move sets during a typical game. In this graph, the line with the diamonds denotes the degree of membership that the start moves exhibit. The line with the boxes shows the membership of the middle moves and the line with the crosses shows the membership of the late moves.

locations and the set of late-moves are a function of the number of discs placed in the edge-locations.

To compute the degree of membership of each of the fuzzy sets start-moves, mid-moves, and late-moves, the three different parametric equations used are Equation 5.3, Equation 5.4 and Equation 5.2, respectively. The result of using the decending S-function (Equation 5.3) for the start-moves set is that with the board in the starting configuration, the degree of membership in the set of start-moves has a value of 1. In this same configuration using the bell S-function (Equation 5.4) for the mid-moves fuzzy set the degree of membership of the mid-moves set is zero and using the decending S-function (Equation 5.3) for the late-moves fuzzy set the degree of membership of the late-moves set is also zero.

As discs are played, the middle locations begin to fill up. As these locations fill, the degree of membership of the start-moves set begins to decline. Any equation that emphasizes the start-moves membership will also decline proportionally. As discs are played in the frontier locations, the degree of membership of the mid-moves begins to increase. This continues until the frontier locations are half occupied. At this point the degree of membership of the mid-moves set begins to decrease until finally when the frontier locations are completely filled the degree of membership once again becomes zero. As occupancy in the edge disc region starts to increase, the degree of membership in the set of late-moves also increases due to the ascending S-function (Equation 5.2) that is used to compute the degree of membership. Ultimately when the edge-disc locations are filled, the degree of membership for late-moves will be 1.

Because it is not required that the mid-locations be completely used before progressing to the frontier-locations or the frontier-locations be completely used before progressing to the edge-locations, it is important that the degree of

membership of the mid-locations, the frontier-locations and the edge-locations all be considered when determining the degree of membership that the current board configuration displays for each of the game stage sets.

Using Zadeh's "compositional rule of inference" for approximate reasoning, Togai and Watanabe utilized multiple *fuzzy variables* in determining a final degree of membership[20]. This concept solved the problem of combining several fuzzy factors to determine a final degree of membership.

The degree of membership in each of the game stage sets is determined by the use of fuzzy variables start-moves, mid-moves and late-moves. By combining these fuzzy variables the degree of membership of each board configuration in each of the game stage sets can be determined. In this research, each of the fuzzy variables are multiplied by a coefficient and added. The coefficients when summed are equal to one. This normalizes the final degree of membership to be $0 \leq f(x) \leq 1$. The maximum of these values determines the game stage that the configuration is in.

The formulas used to compute the degree of membership of the game board configuration in the sets opening game, middle game, and end game are shown in Equation 5.5 - 5.7.

$$\begin{aligned} \text{opening game membership} = \; & 0.6 * \text{start moves membership} + \\ & 0.2 * \text{middle moves membership} + \\ & 0.2 * \text{late moves membership} \end{aligned} \tag{5.5}$$

$$\begin{aligned} \text{middle game membership} = \; & 0.2 * \text{start moves membership} + \\ & 0.6 * \text{middle moves membership} + \\ & 0.2 * \text{late moves membership} \end{aligned} \tag{5.6}$$

$$\begin{aligned} \text{end game membership} = \; & 0.2 * \text{start moves membership} + \\ & 0.2 * \text{middle moves membership} + \\ & 0.6 * \text{late moves membership} \end{aligned} \tag{5.7}$$

The coefficients for these experiments are chosen to be 0.6,0.2 and 0.2. The coefficients of the fuzzy components in Equation 5.5 are set with an emphasis on the type of moves that most typify that stage of the game. It should be apparent that start moves should be more dominant in the opening game than either middle moves or late moves. Similarly in Equation 5.6, the middle moves should dominate, and in Equation 5.7 the late moves should dominate.

Using these equations the graph in Figure 5.6 was generated for the moves in a game where both opponents are determining their strategies based upon the same equations. The game stage can be determined at each move by finding the game stage set with the highest membership value on the graph.

Following the graph from left to right it should be obvious that the game progresses from the opening game through the middle game and finally into the end game.

5.4 Description of Strategies for Each Stage

For this research, a different strategy will be used at each of the three stages of the game.

1. minimum disk strategy - used in the opening game to increase mobility.
2. semi-stable disc - used in middle game to increase the number of semi-stable discs.
3. maximum disk strategy - used in end game to maximize the final disc count.

Minimum square strategy attempts to increase mobility [2]. Although Rosenbloom does not have any direct experience with this strategy, he explains

[2]Recalling chapter 2, mobility is the number of valid moves. Increased mobility is an increased number of valid moves.

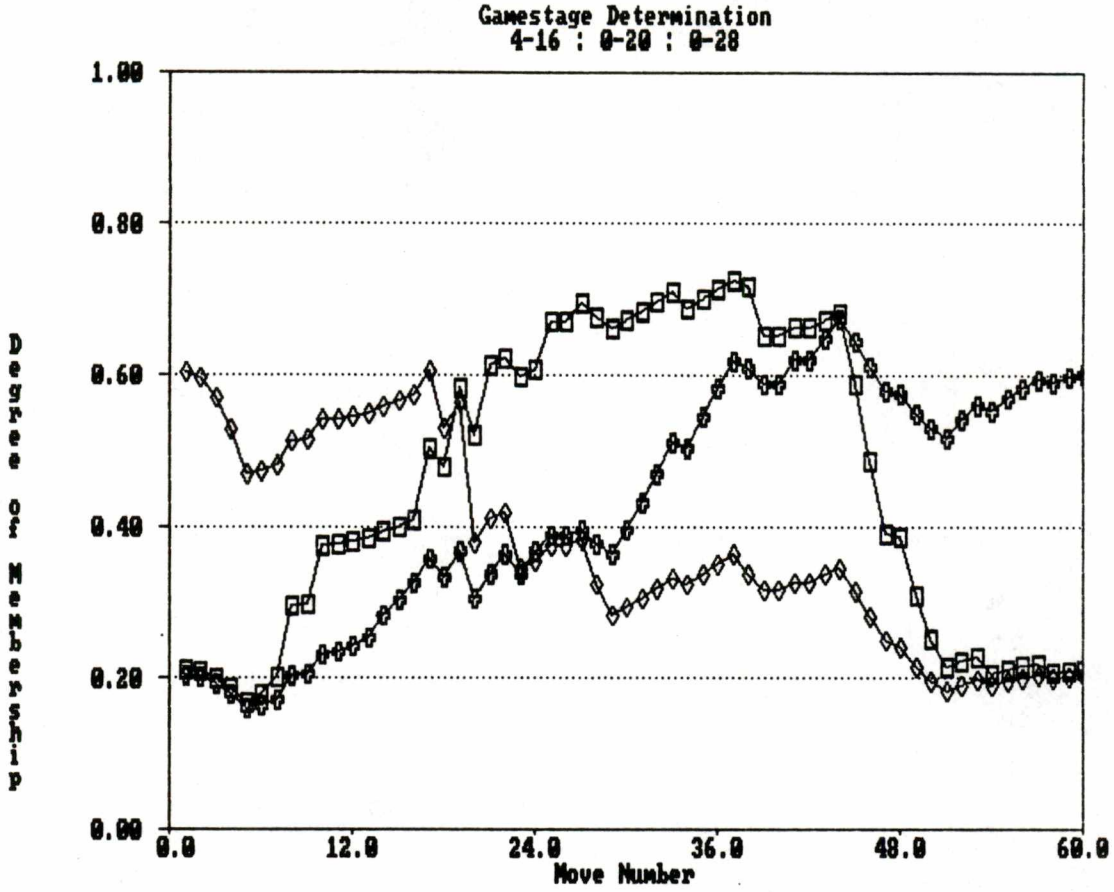


Figure 5.6: Degree of memberships in Game Stage Sets

Figure 5.6 shows the degrees of membership computed for each of the game stage equations (Equations 5.5, 5.6,5.7) during a typical game. In this graph, the line with the diamonds denotes the degree of membership that the opening game exhibits(Equation 5.5). The line with the boxes shows the membership of the middle game (Equation 5.6) and the line with the crosses shows the membership of the end game (Equation 5.7).

his view on the strategy: "... its emphasis is wrong. Where the discs are placed is as crucial as the simple count of how many there are. Having only a few poorly placed discs can be disastrous ..."[16]. Although this may be true, this research does not attempt to define a "best" strategy. Since both computerized players will be using the same strategies, it will not make a difference. An attempt to maximize the mobility in preparation for the middle game strategy, is all that is emphasized.

For the middle game, a strategy is used that attempts to maximize the number of semi-stable discs. A stable disc is a disc that cannot be recaptured for the remainder of the game. A semi-stable disc is a disc that cannot be recaptured on an opponent's next turn. Semi-stable discs are important because often these discs become stable discs.

The end game strategy simply attempts to maximize what is ultimately the final determiner of the game winner, the disc count. If the earlier strategies worked satisfactorily, then this relatively simple strategy should be able to yield a high disc count.

Since this research does not attempt to arbitrate about what strategy is best in what situations, it is only important to note that, like in the weighted square machine learning strategy discussed in Chapter 4, the learning program will only be competing with an opponent using a similar strategy. This way if a strategy is inefficient in any way then it will be equally inefficient for both players.

5.5 Modification of Weighting Coefficients

The coefficients of the fuzzy variables start-moves, mid-moves, and late-moves determine which variable or variables dominate the equation. This determines which stage the game is in and thus which strategy the current board evaluation

should use. Because of the physical characteristics of location occupancy used in this research, the coefficients used simply emphasized the physical characteristic that should be dominant at that stage of the game.

To determine the opening game membership, a coefficient of 0.6 for the start-moves membership is used. The coefficients for both the mid-moves and late-moves variables are both 0.2. This means that to determine the degree of membership in the opening game, the fuzzy variable start-moves is emphasized by a proportion of three to one. To determine the degree of membership in the middle game, the fuzzy variable mid-moves is given a three to one weight. The same was done for the end game emphasizing the late-moves variable. The game stage set with the highest degree of membership in its respective set is used to determine the actual stage that the game is in at that time.

The strategies used at each stage of the game, suggested by the work of Rosenbloom, prove robust enough to defeat any manual [3] attempts to alter the order of the strategies. For instance, an attempt to improve play by using the strategies in different orders proves unproductive.

To cause the strategies to play in a different order, the coefficients are modified for the opening game emphasizing the late-moves variable. The coefficients used to determine the opening game membership are 0.3,0.3 and 0.4 for the start-moves, mid-moves and late-moves respectively. For the middle game, 0.3,0.4,0.3 are used as the coefficients and for the end game 0.4,0.3 and 0.3 are used. This causes the opening game to play the maximum disc strategy and the end game to play the minimum disc strategy. This results in a defeat by the strategy using the original coefficients (0.6, 0.2, 0.2 for the opening game) by a score of 45 discs to zero. It should be noted that the original strategy plays so

[3] "Manual" in this case meaning that for this research the parameters were altered in different ways by the researcher after each experiment. It will be used in contrast to "automatic" where program logic will alter the parameters based on specific rules.

well in comparison to the new strategy, that the entire board is not even filled.

An attempt to manually alter the coefficients to ignore the less important fuzzy variables also fails to improve the quality of play. The coefficient is set to 1 for the start-moves in the opening game and 0 for both mid-moves and late-moves. For the middle game, the coefficient for the mid-moves is set to 1 and both start-moves and late-moves are set to 0. A similar proportion is set for the end game, emphasizing the late-moves. Competing this strategy against the original strategy (0.6, 0.2,0.2), the original strategy wins with a score of 44 discs to zero. Once again the original strategy proves strong enough to defeat the new strategy without filling the board.

In another test, the coefficients of one of the game stage equations are changed. This test uses 0.5, 0.3 and 0.2 for the opening game, 0.2,0.6 and 0.2 for the middle game and 0.2,0.2 and 0.6 for the end game. This strategy also loses to the original strategy of 0.6,0.2 and 0.2 for the opening game.

From the previous experiments, this researcher feels that the coefficients setting is optimal, hence a greater effect on the game performance can be found by altering the parameters that determine the slope of the fuzzy variable graphs. The coefficients 0.6,0.2 and 0.2 for the opening game, 0.2,0.6 and 0.2 for the middle game and 0.2,0.2 and 0.6 for the end game are used throughout the remainder of the experiments involving modifications of membership functions of fuzzy sets.

5.6 Modification of Fuzzy Set Parameters

Since changing the weighting coefficients seem to have no positive effect on improving the performance of the program, other variables must be altered. For this research the parameters that determine the slope of the start-moves, mid-moves and late-moves graphs will be changed so that the game will move

from stage to stage at different times. This researcher considered modifying the parameter determining the degree of fuzziness of the mid-moves function. Changing this though effects both the end of the opening game and the beginning of the end game equally. It is important to consider each of the stage changes separately to determine the best strategy.

The parameters that are altered for this research are the *upper* and *lower* limits of the opening game and end game membership functions. The original parameters are set as the physical upper and lower limits of the amount of discs in each region. Referring back to Figure 5.4 shows the number of locations in each region. The upper and lower bounds for the opening game are set to 4 and 16 respectively. For the middle game, they are set to 0 and 20 and for the end game 0 and 28. In a typical game, these values create the graph shown in Figure 5.6.

In an effort to improve play, the parameters are changed by the programmer. These parameters are changed to determine whether these changes would effect the way the program played. The lower bound of the start-moves is left at 4 but the upper bound is changed to 8 (half of the total locations). The mid-moves bounds are left as before. The late-moves lower bound is changed to 14 (once again half the total locations) and the upper bound is left as before at 28.

Using these parameters, a typical game produces the graph in Figure 5.7.

Comparing Figure 5.6 with Figure 5.7 will show that with the modified parameters the game begins playing the middle game strategy sooner than with the original parameters. These parameters also cause the program to stop using the middle game strategy sooner and begin using the end game strategy.

Although these changes may seem insignificant, playing the new strategy against the original strategy resulted in a decisive victory for the new strategy.

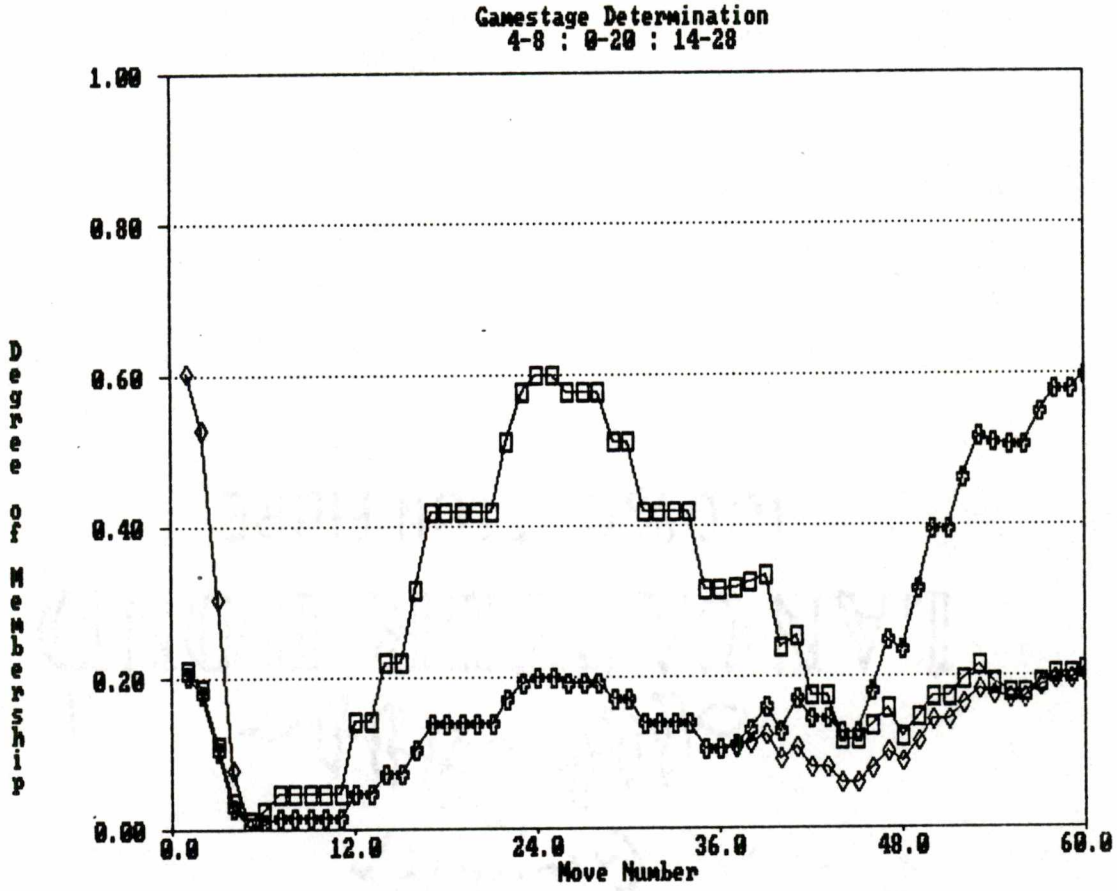


Figure 5.7: Degree of memberships in game stage sets with modified parameters

Figure 5.7 shows the functions produced when the start-moves bounds were at 4 and 8, the mid-moves bounds at 0 and 20 and the late-moves bounds at 14 and 28.

Figure 5.8 shows the disc difference after each turn in the game. The disc difference is found by subtracting the number of discs owned by the original strategy at each turn from the discs controlled by the new strategy.

5.7 Using Machine Learning to Modify the Parameters

With it shown that changing the parameters can improve how a game program plays, modifying the program to *automatically* adjust these parameters was the next phase of the research.

Starting with the original strategy parameters set at the physical limits, the upper limit of the start moves membership affecting when the opening game concludes is reduced so that the opening game ends earlier in the game. The middle game strategy then begins sooner.

The program is modified so that both the "student" and "expert" begins with the parameters set to the physical limits. Before playing each of the individual games, the upper bound of the start moves fuzzy membership function is reduced by one. The game is then played against the expert strategy and the student's new strategy.

At the conclusion of the game, if the student is the winner then the student's new strategy replaces the expert strategy. If the expert wins then a counter, indicating how many consecutive wins by the expert, is incremented. If the counter reaches five (five successive games that the student did not show a winning improvement) then the program terminates[4].

The computerized opponents continue to play against one another as long as the upper bound of the start moves is greater than or equal to the lower bound

[4]In these experiments the changes made always improved the play of the student so this option is never utilized.

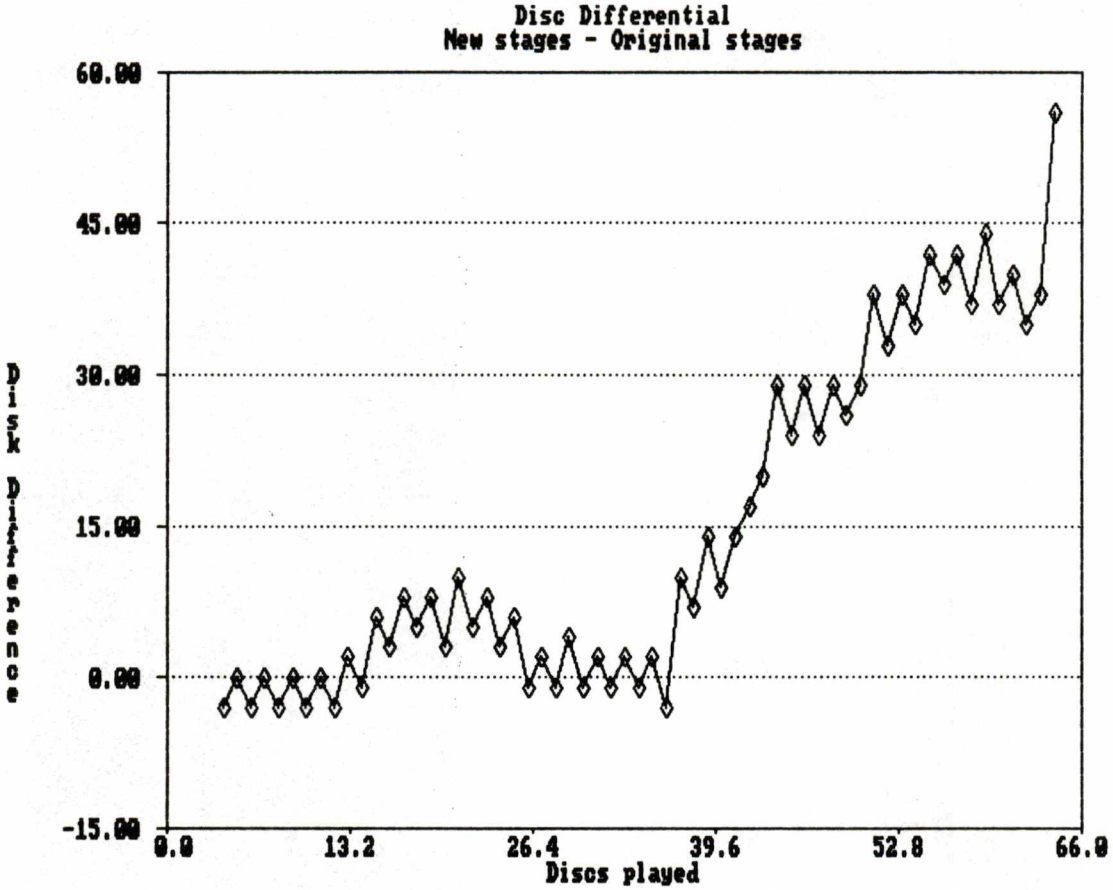


Figure 5.8: Disc differences between original strategy and new strategy

This graph shows the disc differences when competing the original strategy with the new modified strategy. The difference is found by subtracting the number of discs owned by the original strategy from the discs controlled by the new strategy.

of the start moves. The expert won the first three games. Because of this the expert strategy remained at the physical upper limit (16). At the conclusion of the fourth game though the student won. At this point the student's strategy replaced the expert's strategy.

The games continued to play in a similar manner. In this experiment the strategy created by reducing the upper bound of the start moves proved stronger than the previous strategy. Because of this the student's strategy always replaced the expert's strategy until the upper bound was reduced to be equal to the lower bound. This shows that each modification improved play.

After these games were played, the curves showing the game stage during the final game was plotted. This graph is shown in Figure 5.9.

This experiment demonstrates that the strategy used in the opening game can not compete favorably against the stronger middle game strategy. Although it is not the results that were expected, these results do show improved play.

In an attempt to further improve play the learning strategy is changed somewhat. Since the upper bound of the opening game was modified before and showed improved performance, the new strategy was used as the expert from which to begin modifying the end game parameters. The lower bound of the end moves fuzzy set is increased to cause the end game to begin later in the game.

The lower bound of the end moves in increased for the student at the beginning of each game. If the student wins the game then the experts parameters are replaced by that of the student. If the expert wins then the expert strategy remains the same.

As happened modifying the start moves parameters, the strategy used in the middle game proved dominant. The student strategy continued to win every game thus modifying the expert strategy every game. When the lower bound of the late moves became equal to the upper bound of the late moves,

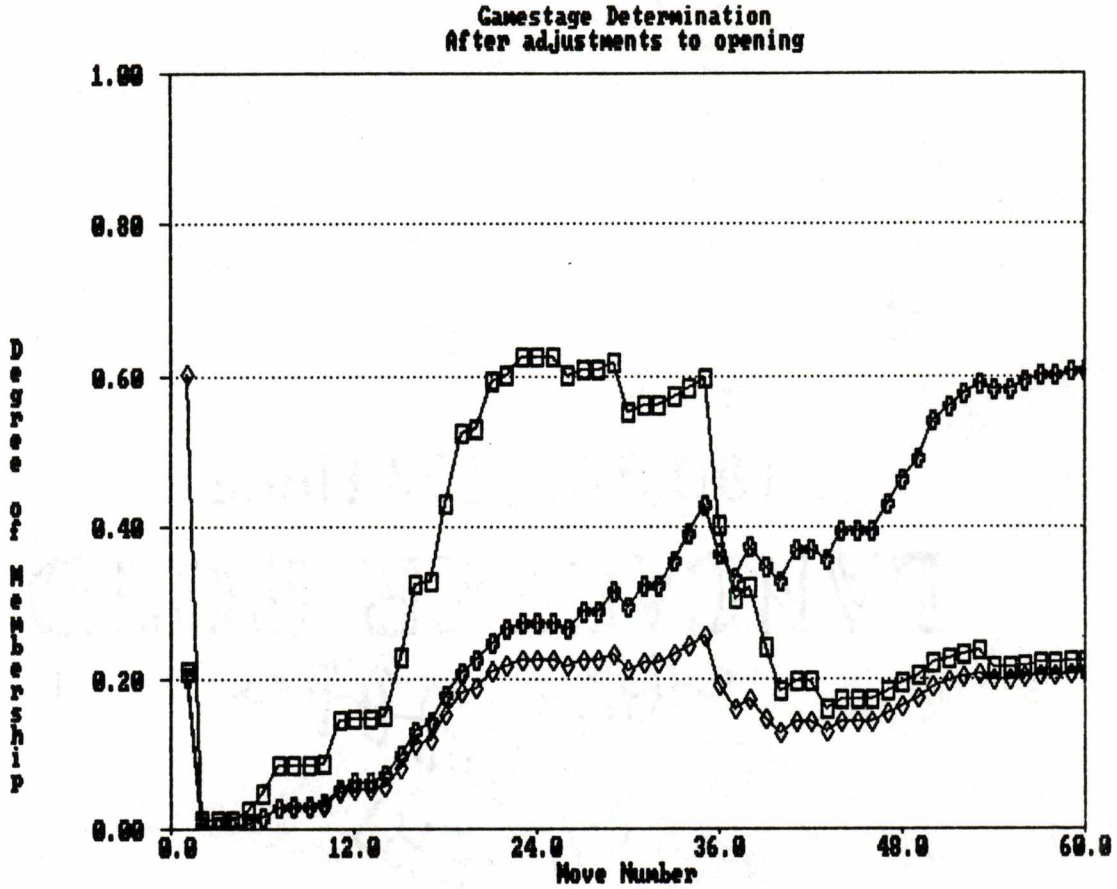


Figure 5.9: Game stage sets with modified start moves parameters

Figure 5.9 shows the functions produced by the student strategy after the learning phase affecting the end of the opening game strategy.

then obviously play had to stop.

The graph shown in Figure 5.10 shows the degree of membership in each of the game stage sets when the strategy modified by the computer is played against an opponent playing the same strategy.

To really show that the play has improved, the final strategy created by both phases of learning was played against the original strategy. It should be remembered that the original strategy has its lower and upper bounds set equal to the physical upper and lower numbers of locations in each region.

The results of the game showed that the new strategy was able to beat the original strategy with a score of 38 for the new strategy and 26 for the original strategy. The goal of improved play had been achieved.

Experiments were performed to confirm these results. Instead of making the opening game end sooner, the equations were modified to make the opening game play longer. To do this a process called *dilation*[25] is employed to cause the fuzzy variables start-moves and late-moves to have more effect toward the middle of the game.

To do this, the degree of membership for each of the fuzzy variables is computed as in Equations 5.5, 5.6,5.7. This degree of membership is taken to some power $\frac{1}{n}$ (where n is any real number greater than 0) to determine DIL($M(u)$) or the dilation of $M(u)$. For example, if the degree of membership M for some value u is computed then a formula such as Equation 5.8 would be used to dilate it [25].

$$\text{DIL}(M(u)) = M(u)^{\frac{1}{n}} \tag{5.8}$$

The degree of membership for the mid-moves was left at its original values. The degree of memberships for the start-moves and late moves were each

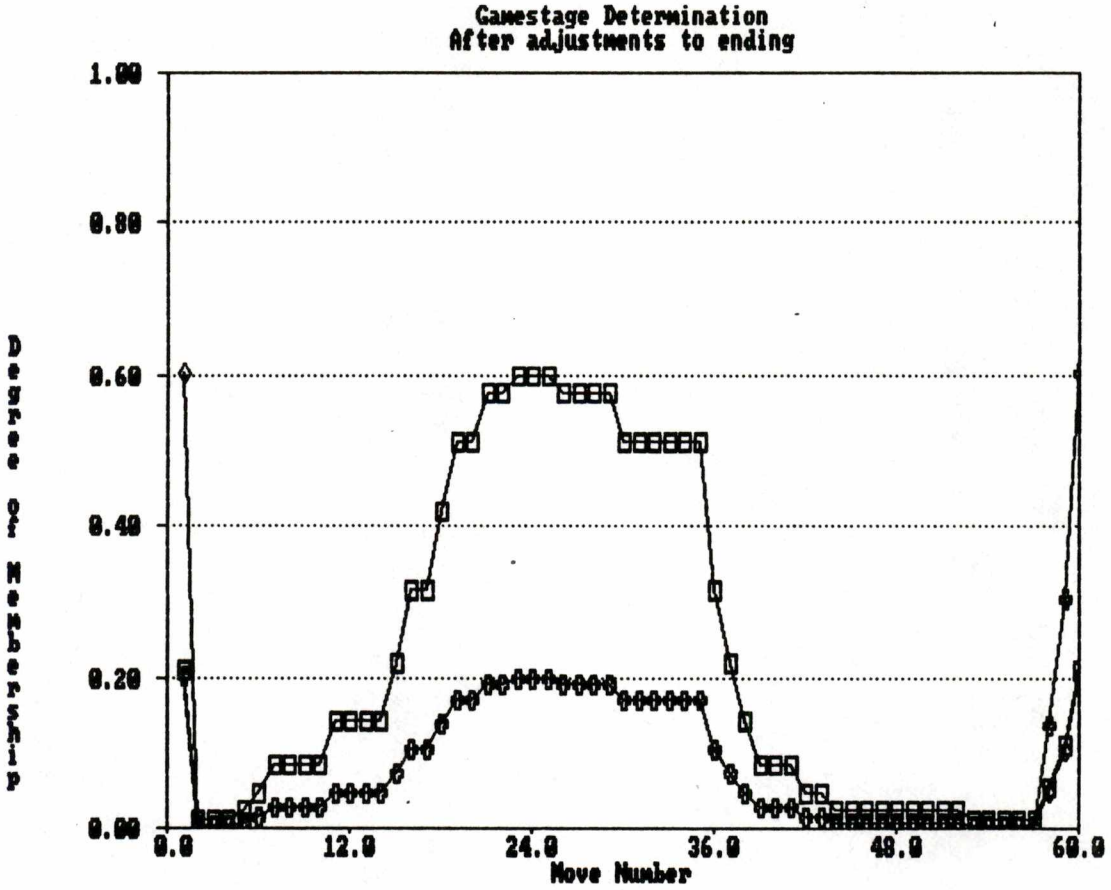


Figure 5.10: Game stage sets with modified late moves parameters

Figure 5.10 shows the functions produced by the student strategy after the learning phase affecting the start of the end game strategy.

dilated so that the opening game would proceed longer and the end game would begin sooner. An example of how this graph looks with n equal to 4 is shown in Figure 5.11.

To see if dilation would improve the quality of play, values for n were manually set. In the first attempt, n was set to 2 for both the start-moves variable and the late-moves variable so that the square root of the membership would be taken. This was played against the original strategy that has its parameters set at the physical limits of each of the regions. This resulted in a victory for the original strategy of 51 discs to the new strategies 13 discs. The value of n continued to be increased manually to 3 then to 4 and to 5, each with a similar victory for the original strategy.

These experiments showed what was suspected after the the tests where the bounds were automatically changed. The program determined the best time to go from game stage to game stage. This demonstrates the fact that the strategy used in the middle game proves so overpowering that the opening and the end game strategies can not compete favorably. It is unfortunate that the equilibrium for the game stage classification could not be found where each of the strategies contributed to the overall strategy for the game.

5.8 Conclusions of Game Stage Modification

It is important to point out that the techniques used here did find the best times to change stages based upon the strategies used in each stage. If a stronger opening game strategy could be identified, then perhaps this technique could be shown to be somewhat more useful for Othello.

In the experiments affecting the lower bounds of the end moves, although the student continued to win until the lower bound became equal to the upper

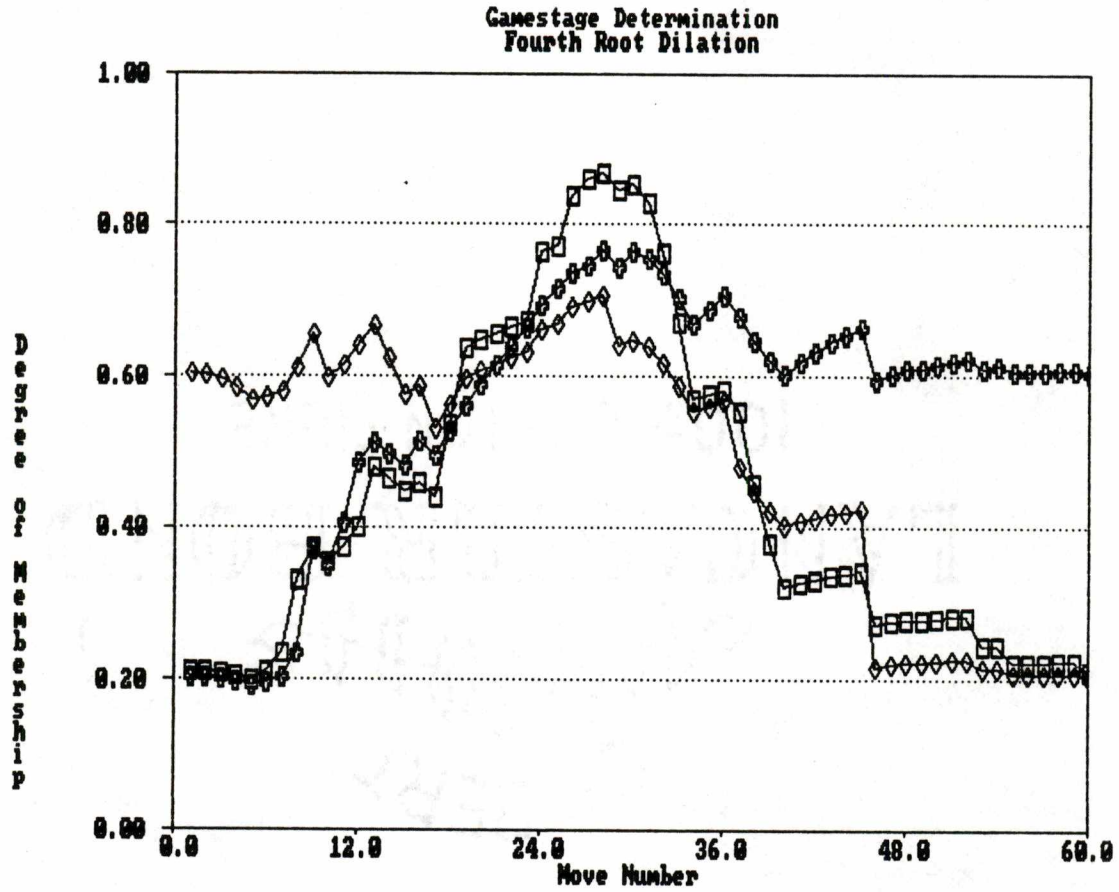


Figure 5.11: Degree of memberships in Game Stage Sets using Dilation

Figure 5.11 shows the degrees of membership that each of the regions demonstrate in the game stage sets during a typical game using dilation of the start-moves and late-moves fuzzy variables. In this graph, the line with the diamonds denotes the degree of membership that the opening game exhibits. The line with the boxes shows the membership of the middle game and the line with the crosses shows the membership of the end game.

bound, the margin of victory for the student remained reasonably constant for the last eight or so games. This indicates that the difference may not be significant between each of the tested strategies toward the end of the run of games.

Modification of the parameters using machine learning did improve the play. The strategies used for each of the game stages probably contributed to the less than optimum results produced by the program. By using stronger strategies, the equilibrium of the game stages may balance at a point where all stages of the game contribute to the overall game strategy.

Chapter 6

CONCLUSIONS

A maximum disc strategy would work for Othello if the entire game tree could be evaluated at each move. Since this is impossible, other characteristics of a winning game must be be evaluated to determine the quality of a game board configuration at any time during the game. Honing the evaluation of these characteristics to more accurately reflect the actual quality of the game board configuration is a problem that this researcher has attempted to solve.

6.1 Success of Research

As was shown in the weighted square research, complicated strategies are not needed for a program to play reasonably well. The square weightings provide a good method for choosing the next move in a game tree.

Machine learning is shown to be an effective tool to improve the performance of a game playing program. This research shows that after repeated games, an uninformed strategy can play effectively against an informed strategy. This is true even with a simple game playing strategy.

Using a low threshold of acceptance produced only marginally effective results. The uninformed strategy was easily misled, placing undue importance on squares that were not indicative of a classically well played game. Using a

threshold of acceptance improved the quality of play. This showed that even using a measure such as a "won game" as the motive for changing the value of squares, the quality of play could be improved.

Since, in this case the mutations proved lethal to the winning strategy, other more complex mutation algorithms may need to be investigated. An area for future research may be to have the mutation algorithm behave less severely toward the student when the student wins. Another approach may be that if the student wins by a very large margin then the student has learned all that it can from that expert and a new expert should be chosen.

A much more interesting and effective strategy for playing Othello was utilized in the second phase of this thesis. It is impractical to treat situations found at the beginning of a game with the same strategy as situations found at the end of the game. Because of this a study of the game stage classification proved to be most interesting.

The area of game stage classification is complicated by its nature but is shown in this research to very much fall into the area of fuzzy set theory. As with anything quantifiable, heuristics must be selected that describe the physical situation being quantified. This research chose very simple physical characteristics upon which to base the quantification. These regions of differing locations on the game board provide an easily countable physical characteristic. Even with these superficial characteristics, improved play was shown.

When the research was undertaken to adjust the game stage parameters, it was expected that a more reasonable equilibrium would be reached. As it turns out, the upper bound of the start moves membership function continued to be reduced until it became equal to the lower bound of that function. Conversely when the late moves parameters were adjusted, the lower bound was increased until it became equal to the upper bound of that function. Improved play was

shown but due to the strength of the middle game strategy, the other strategies could not compete.

6.2 Application of Research Outside Game Theory

Many expert systems and artificial intelligence programs rely heavily upon the quantification of characteristics of a physical situation. Many times, the method of making a recommendation is based upon these physical attributes and the recommendation will be different based upon other characteristics. The boundary between these times may be unknown or decided by an *educated guess.* Using machine learning to modify the importance of these characteristics can play a major part in honing these parameters to their best possible values.

This may be applied to several areas. An example of this may be the following:

Suppose a service technician was required to make a decision as to whether to repair or replace a computer peripheral such as a hard disk drive. There are several factors that determine his recommended procedure.

The technician's first decision to make is whether the disk is repairable or not. Since an unrepairable drive must be replaced, it will be assumed that the disk is repairable. The harder question is whether or not it is cost effective to repair a drive. For this example, two sets, 1) cost effective and 2) not cost effective will be defined. "Cost effective" disk drives will be repaired and "non cost effective" drives will be replaced with a new model (probably at some cost to the end user).

The determination of a drives cost effectiveness is more then just a matter of months since the drive was installed. Factors include:

- The measured or calculated number of hours that the disk has been in use.

- The physical distance from the nearest service technician.
- The physical distance from the spare parts warehouse [1] .
- The environment in which the drive is normally used including:
    1. The temperature differences between night and day and summer and winter.
    2. The particulate matter in the air.
- The expected cost of the actual repair.
- The expected time to the next failure.

It should be obvious that as the number of hours of operation increases, the greater the likelihood that the drive will be replaced. This may not be linear because many factory warranties are prorated. The closer the drive is to being out of warranty, the more the factory will charge to replace the drive. A service technician within a reasonable distance to the location of the drive, will decrease the chances of replacing the drive. Although the amount of time for the technician to repair the drive would be the same, the travel costs of a local technician would be greatly reduced. The distance from the parts warehouse is a function that increases for from a short to medium distance and decreases from a medium to large distance. Why is this? Air freight for shipping parts is of a reasonable cost from major city to major city but becomes greater from a rural city to another rural city.

By plotting these and other costs associated with repairing computer peripherals, these functions can be combined (initially based upon an "educated

[1] A distinction between the parts warehouse and service technician must be made because some repairs do not require new parts. The down time for getting replacement parts must be considered when "aging" a drive.

guess") to produce a degree of membership in each of the sets *cost effective* and *non cost effective.*

How can these functions be modified to reflect the most profitable situation? By comparing costs, either with past statistical data, projected statistical data or experimental test data, different solutions can be proposed and evaluated by a service department. As cheaper solutions are found, these parameters are kept and tests continue to be run.

Eventually the most cost effective method of determining repair or replacement would be found based on the observable and quantifiable parameters. It is important to note that there can be as many fuzzy variables as is necessary to describe the physical situation. It may be time consuming to analyze all of the data and adjust parameters and coefficients but the time spent would pay off in a much clearer method of deciding whether or not a disk drive should be replaced.

As the above example shows, the technique of modifying fuzzy set coefficients and parameters is by no means limited to the game theory. This technique can have applications in any field where importance of different characteristics are not easily be quantifiable as *true* or *false* and the parameters of the ensuing fuzzy set are uncertain.

6.3 Areas of Continued Research

Othello is a largely unexplored area of game research. This is undoubtedly due to the fact that few people in the United States play it. Research of such topics as game theory, pattern recognition, machine learning and, as is shown in this research, application of fuzzy set theory can easily be applied to games such as Othello.

Some areas of Othello not studied in this paper may be viable research

topics. Although the weighted square research may not lead to championship strategy, it provides a excellent arena in which to study the modification of parameters of a situation. The "mutations" that were used to prevent stagnation could be modified to be more selective in their effect. Preventing the mutations from destroying a strategy that is winning would probably be the best first step. Modifying the way that the weights are increased after each game might also be an area of future research. It may be useful to modify the weights of each square proportionally to the margin of the winning disc difference. This proportion would would cause the learning program to gain knowledge in proportion to how good its opponent is.

Even modifying the way that each game board configuration is evaluated may of interest. In this thesis, to determine the quality of a proposed move using weighted squares, the disc values of the first player are subtracted from the disc values of their opponent to provide a final evaluation. Instead of this additive evaluation of a move, perhaps a proportion between the the two player's disc counts in the proposed configuration would provide a better measure for comparing proposed moves.

The game stage classification is definitely the most interesting (and this researcher feels that it is the most promising) area of research into the game of Othello. It may be useful to use parameters that are not so closely linked with each of the actual game stages. For instance, identifying other characteristics to use as the fuzzy variables may prove beneficial. Examples would be characteristics such as the number of static discs, quality of moves available for each player based on a weighted square matrix, or mobility[19].

The strategies used in the opening and end game stages could be improved to reach a more even balance and allow all of the game stage strategies to be fully utilized. A simple minimum disc strategy was used in the opening game.

Perhaps as Rosenbloom states, "Where the discs are placed is as crucial as the simple count of how many there are." Future research may want not to discount this statement.

It would be important to identify stronger strategies for each of the game stage classifications. Machine learning can be used to determine if in actuality these strategies are good for that stage of the game.

Going beyond the chronological game stage idea, fuzzy sets such as *threatening* or *being threatened* could be defined. Characteristics that describe each of these sets could be used as fuzzy variables thus defining the evaluation and thus the strategy. Ideally a program that would create these strategies would demonstrate true machine learning. Perhaps these techniques of automatically modifying the parameters of a fuzzy set function will contribute to this goal.

Another variation of this technique would use the game stage evaluations as coefficients of each stage evaluation to determine the final evaluation of the board configuration. For example assume that the current game board configuration has a 0.264 degree of membership in the opening game and the opening game and the opening game evaluation is 167. Also assume that the game board has a membership of 0.623 in the middle game and the middle game evaluation is 103. The end game has a membership of 0.142 and and evaluation of 24. Multiplying the memberships by their respective evaluations would generate a final evaluation. This evaluation would be the relative goodness of the game board.

To use a static strategy and hard sets from beginning to end may work reasonably well for some game theory situations but may not translate well into an actual application. By modifying fuzzy variables, a strategy or series of strategies can be modified to produce an optimum playing strategy based on observable characteristics.

Fuzzy sets in games or in practical work by their very nature have uncertain parameters. Using machine learning to modify the parameters of the fuzzy sets can bring those parameters to their best possible values. This can be used to produce the best fuzzy set parameters.

BIBLIOGRAPHY

Bibliography

[1] Arnold, P. *Illustrated Book of Table Games*, St. Martin's Press, 1975

[2] Abramson, B., R. E. Korf. A Model of Two-Player Evaluation Functions, *Automated Reasoning*, pp. 90–94, 1987

[3] Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Publishing Corp.,1981

[4] Bowden, B.B. *Faster than Thought*, Pitman, London, 1953.

[5] Brown, D.J.H., and S. Dowsey. The Challenge of Go,*New Sci.*, vol. 81, 1979

[6] Coombs, C.,R. Dawes, A. Tversky. *Mathematical Psychology*,Prentice-Hall, Inc., Englewood Cliffs, NJ, 1970

[7] DeJong, K.A., A. C. Schultz. Using Experience-Based Learning in Game Playing, to appear in the Proceeding of the Fifth Conference on Machine Learning, Jun. 1988

[8] Dubois, D., H. Prade. *Fuzzy Sets and Systems*, Academic Press, Inc., New York, 1980

[9] Gabriel. *Othello Rule Book*, Kohner, Inc., Elmwood Park, NJ, 1977

[10] Hoffman, P. *MEL A Learning Program that Improves by Experience in Playing the Game of MILL*, University of Illinois at Urbana-Champaign, Jan. 1982

[11] Kahneman, Slovic, Tversky. *Judgement Under Uncertainty: Heuristics and Biases*, Cambridge University Press, 1982

[12] Liepins, G. E., M. R. Hilliard, J. Richardson, M. Palmer, Genetic Algorithms as Pseudo-Random Search Operators, University of Tennessee, to be published, 1988

[13] Reinke, R.E. Knowledge Acquisition and Refinement Tools for the ADVISE Meta-Expert System, M.S. Thesis, Department of Computer Science, University of Illinois, 1984

[14] Reitman, W., B. Wilcox. The Structure and Performance of the INTERIM.2 GO Program, In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* Vol. 2, sponsored by International Joint Conferences on Artificial Intelligence, Inc. Information Processing Society of Japan, Tokyo, 1979

[15] Rich, E. *Artificial Intelligence*, McGraw-Hill Book Company, 1983

[16] Rosenbloom,P.S. A World-Championship-Level Othello Program, *Artificial Intelligence,* vol. 19,pp. 279-320, 1982

[17] Samuel,A. Some studies in Machine Learning Using the Game of Checkers,*IBM Systems Journal,* vol. 3, 211-259, 1959, Reprinted and cited from *Computers and Thought*, E. Feigenbaum, J. Feldman, Krieger Publishing Malabar Florida,1981

[18] Samuel,A. Some studies in Machine Learning Using the Game of Checkers II –Recent Progress, *IBM Journal of Research and Development*, vol. 11, November 1967

[19] Teague, A. H., Private correspondence, June 1988

[20] Togai, M., H. Watanabe *Expert System on a chip: An Engine for Real-Time Approximate Reasoning*, "Proceedings of the ACM SIGART International Symposium on Methodologies for Intelligent Systems", Ras, Z., M. Zemankova, 1986

[21] Von Neumann,J., O. Morgenstern, *Theory of Games and Economic Behavior*, J. Wiley and Sons, New York 1964

[22] Zadeh, L.A. *Fuzzy Sets*, Information and Control, vol. 8, 1965 pp. 338-353

[23] Zadeh, L. A. *Quantitative Fuzzy Semantics*, Information Sciences, vol. 3, 1971, pp.159-176

[24] Zadeh, L. A. *Fuzzy Logic and Approximate Reasoning*, Memorandum No. ERL-M411, Electronic Research Lab., University of California, Berkley, 12 Nov. 1974

[25] Zemankova, M. *Fuzzy Relational Databases*, Doctoral dissertation, Florida State University, Dec. 1983

VITA

Lawrence R. Webb was born in Kingsport, Tennessee on September 1, 1960. He attended elementary school in that city and graduated from Dobyns-Bennett High School in June 1978. The following September he entered East Tennessee State University, and in June 1982 he received a Bachelor of Science degree in Biology.

After working with Renlar Systems in Lexington, Kentucky from June 1982 until September 1985 as an applications programmer, he was accepted to the Master's program in computer science at the University of Tennessee.

While doing graduate work, he accepted a graduate teaching assistanceship in the department. He was married in October of 1986. The author received a Master of Science degree from the University of Tennessee in August of 1988.