# Collaborative Computer Personalities in the Game of Chess

By

Aidan Haran B.Sc.

**A Dissertation Presented in Fulfilment of the Requirements for the M.Sc. Degree**

Supervisor: Dr. David Sinclair

School of Computer Applications
DCU

August 2002

# Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Masters in Computer Applications is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Aidan Mann

ID No.: 95069861

Date: 16/8/2002.

# Acknowledgements

I would like to thank my supervisor, Dr David Sinclair, for his help and guidance during my research and write-up.

Thanks to all my fellow post-grads who make the time pass so quickly especially Paul Browne, Kieran McDonald, Jer Hayes, Thomas Sødring, Jiamin Ye, Cathal Gurrin, Dr Wu Hai and Dr Hyowon Lee. Best of luck to all of you who are still working on your PhD degrees.

I would especially like to thank my parents and family for their support and assistance during my extended stay in college. Thanks!

# Table of Contents

# Abstract

Computer chess has played a crucial role in Artificial Intelligence research since the creation of the modern computer. It has gained this prominent position due to the large domain that it encompasses, including psychology, philosophy and computer science. The new and innovative techniques initially created for computer chess have often been successfully transferred to other divergent research areas such as theorem provers and economic models. The progress achieved by computers in the game of chess has been illustrated by Deep Blue's famous victory over Garry Kasparov in 1997. However, further improvements are required if more complex problems are to be solved.

In 1999 the Kasparov versus the World match took place over the Internet. The match allowed chess players from around the world to collaborate in a single game of chess against the then world champion, Garry Kasparov. The game was closely fought with Kasparov coming out on top. One of the most surprising aspects of the contest was the high quality of play achieved by the World team. The World team consisted of players with varying skill and style of play, despite this they achieved a level of play that was considered better than any of its individual members. The purpose of this research is to investigate if collaboration by different players can be successfully transferred to the domain of computer chess.

# Chapter 1

# Introduction

## 1.1 Importance of Chess in Artificial Intelligence Research

Artificial Intelligence, simply referred to as AI, is the area of research concerned with the creation of computational entities that exhibit intelligent behaviour. These entities would have the ability to perform difficult or unwanted tasks on our behalf. Since the inception of AI, fantastical ideas of the progress of AI have been made, such as the character of HAL in the film "2001: A Space Odyssey". Unfortunately science fiction left reality far behind in the creation of intelligent entities. Progress in AI has been slow and the benefits from its progress are not easily distinguishable. Chess, however, is one area where significant progress has been made and where progress can be easily measured.

AI encompasses many different research fields including computer science, psychology and philosophy. Chess has been central to much AI research because of its crossover into all of these areas. Among the reasons for chess's place of prominence in AI research are [Uiterwijk 1995]:

1. Chess has always been viewed as a game that requires intelligence to play. For this reason chess has been of particular interest to psychologists. By understanding human chess play, one could also extract understanding of human intelligence in general. Chess also has the added benefit that it

is a game that exists in its own little microcosm of reality. The consequence of this is less interference from outside influences and the reduction of data to be observed to the pieces on the game board and the rules of the game.

2. For engineers wishing to create an intelligent machine, chess is a perfect vehicle for their research. Because of the assumption of intelligence to play chess an engineer can avoid the difficult, and unresolved, philosophical questions associated with intelligence. The intelligence of a machine can be proved if it performs a task for which intelligence is assumed. For this reason computer chess has been a popular endeavour for engineers wishing to create intelligent machines.

3. Games are suitable for AI research as their aims are generally clearly defined and their progress is easy to measure. Because their aims are clearly defined it is possible to easily tell if the program is showing signs of intelligence or not. The quality of play in games can be easily determined. In the case of poker, progress can be measured by the amount of money won and in chess by the number of games won, lost and drawn. Chess has an advantage over other games due to the presence of the ELO rating system [Elo 1978] in chess. The ELO rating system, created by the mathematician Arpad Elo, gives chess players a rating based on their previous matches. Based on their ELO ratings the expected result from two players competing against each other can be calculated. If a player achieves a better result than expected their rating increases, if they

achieve a worse result their rating decreases. The mathematical nature of the rating system ensures its scientific usability. The ELO rating gives a standard method of comparing the quality of computer chess programs and allows the progress of computer chess to be compared against that of humans.

Alan Turing, a British mathematician and pioneer in artificial intelligence, wrote the first computer chess program in 1950 [Turing 1950]. At the time the execution of his program had to be simulated using paper and pencil. The program only considered one move ahead into a game. The program played terrible chess but proved that computers could play the game of chess. In 1997 the chess machine Deep Blue [Deep Blue URL] defeated Garry Kasparov, the then World Champion, in a 6 game match. This was a landmark achievement and one that demonstrated the progress made by both computer chess and AI.

## 1.2 Purpose of Research

Deep Blue's victory over Kasparov was a great achievement but there is still a lot of room for improvement. Deep Blue was a massively parallel, special-purpose machine created for the sole purpose of playing chess. When complete, it had cost several million of dollars and taken over five years to create. This type of investment of money and time is impractical for everyday tasks. Computer chess has always been a benchmark for AI research rather than the final goal. Other similar problems such as the Japanese game Go, economic models and medical simulations require much more computational power than chess and advances in computer chess bring the solutions to these problems closer.

In most chess positions it is impractical to search all the possible games that could follow. Therefore, chess programs search only a number of moves forward into the games. The positions at the end of these searches are called terminal positions. An evaluation function is used to assign a score to these terminal positions, which estimates the value of the position for a player (see Section 2.2). Modern computer chess programs use the paradigm that there is a single evaluation function that results in "optimum play". The purpose of this work is to present my research concerning the possible use of multiple evaluation functions to achieve better chess play.

To set the context for this research the methods and ideas currently used in computer chess are presented in Chapter 2. The chapter concludes with problems associated with current computer chess technologies. In Chapter 3 the Kasparov versus the World match is presented. The match was the inspiration for this research and suggested that multiple players collaborating could achieve better play than any of its members individually. The adaptation of this conclusion for computer chess is presented in Chapter 4 along with the limitations imposed on any system by current chess technologies. The test system created to test the research is presented in Chapter 5. Chapter 6 describes the tests performed, the results and factors that influenced the results of tests. Finally in the conclusions section of this work, Chapter 7, the meaning of the results is discussed along with further research that could be performed.

# Chapter 2

# How Computers Play Chess

## 2.1 Viewing Chess as a Game Tree

The game of chess involves two opponents, called *sides*, of different colour, generally white and black, who battle on an 8x8 chessboard by moving their pieces around the board. Chess rules (Appendix A) govern how pieces can move around the board and capture enemy pieces. The purpose of the game is to capture the opponent's king piece first. A game of chess begins with white to move and the board in the *starting position*, which is as follows (see Appendix B for chess notation):

**White:** Ra1, Nb1, Bc1, Qd1, Ke1, Bf1, Ng1, Rh1, and Pa2… Ph2

**Black:** Ra8, Nb8, Bc8, Qd8, Ke8, Bf8, Ng8, Rh8, and Pa7… Ph7

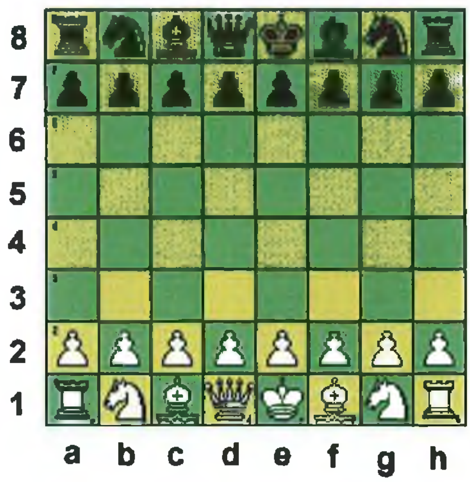



**Figure 2.1:** Chess board in starting position.

Turn to move is alternated between the two sides as moves are made. When a side has the turn to move it must make a move if possible, a side cannot skip a go. If a legal move cannot be made by the side to move while its king piece is in check then that side loses the game. Alternatively, if the side to move cannot make a legal move while its king piece is not in check then the game is declared a draw.

In computer chess a move by either side is called a ply. This is to differentiate from human chess players meaning of the word "move", which is the combination of a move by each player.

From the starting position white can choose a move from 20 different possibilities (a2-a3… h2-h3, a2-a4… h2-h4, Nb1-a3, Nb1-c3, Ng1-f3, Ng1-h3). Following white's move, regardless of the move it played, black can respond by playing any of 20 different moves (a7-a6… h7-h6, a7-a5… h7-h5, Nb8-a6, Nb8-c6, Ng8-f6, Ng8-h6). Therefore, after the first 2-ply of a game of chess the board can be in any of 400 different positions.

The first 2-ply of the game of chess can be represented using the simple tree structure given in Figure 2.2. This type of tree structure can be used to describe many types of games and is called a *game tree*. The nodes of the game tree represent board positions and the edges, the lines connecting the various nodes, represent legal moves that can be played from that position. The progress of the game flows from the root of the tree, which is the starting position, downwards.

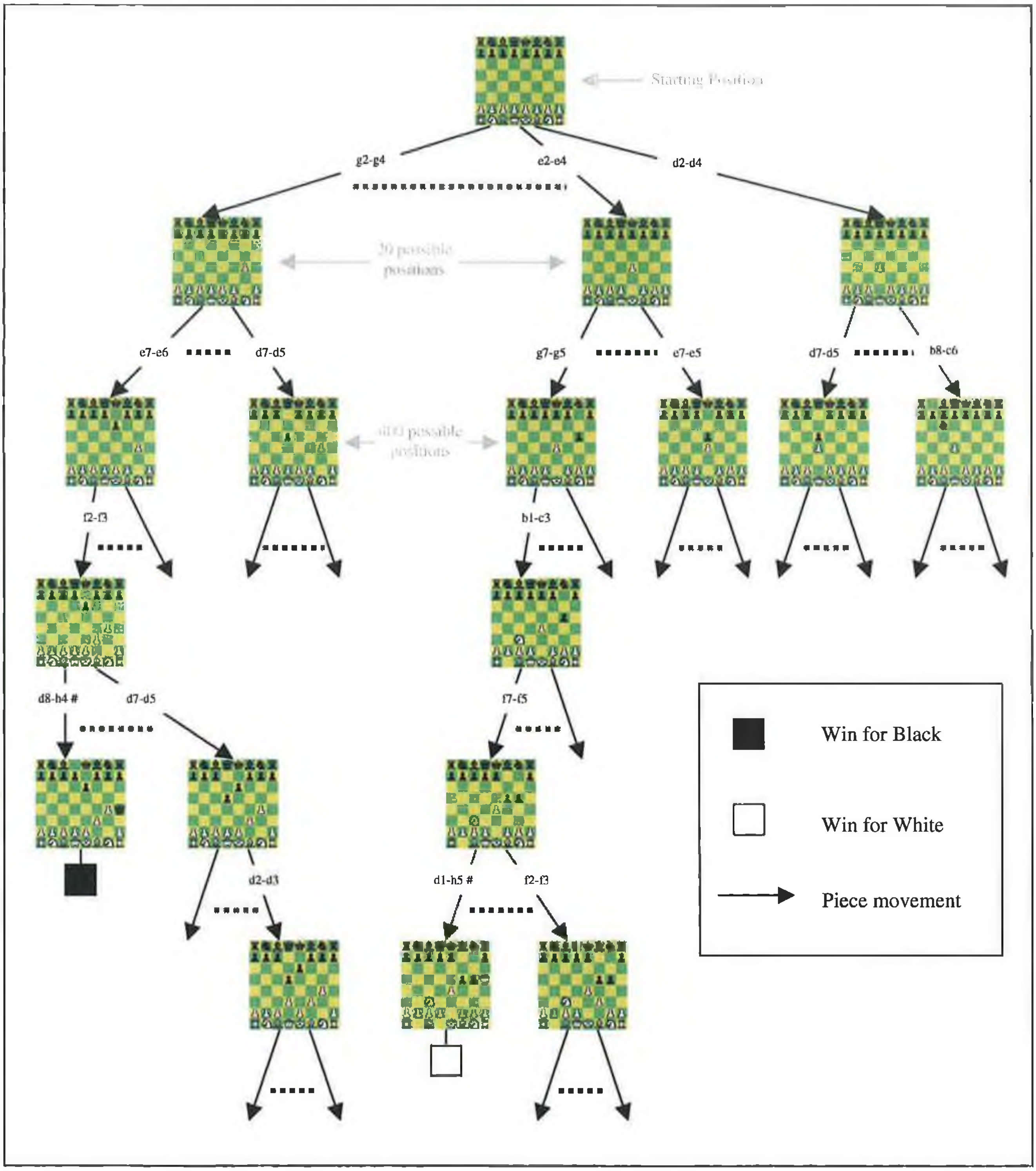



**Figure 2.2:** Chess game tree expanded.

A node in a game tree can be expanded if the node is not a game theoretical terminal where the value of the game has been determined (whether the game is a win, loss or draw). Expanding a node will add a new level below containing the positions that succeed it. In Figure 2.2 each of the positions at the bottom of the tree could be expanded since none of them are game theoretical terminals. Their successors could then be expanded and so on. By fully expanding levels of the game tree the number of nodes in the game tree grows exponentially. This exponential rate of growth increases and decreases as one progresses through the game. At the starting position the exponential rate of growth is 20. This rate increases as more pieces come into play and the rate decreases as pieces are captured and trapped.

Chess is a game of *perfect information*. In a game of perfect information the full states of all game situations are completely visible, the game does not have any elements of chance. Consequently, if one could expand the entire game tree one could achieve perfect play that ensures the best possible result from a given position.

Unfortunately, the game tree of chess is far too large to be fully expanded. The average length of a professional chess game is around 50 moves and the average exponential game tree growth is 35. The number of legal chess games has been estimated at around $10^{44}$, a number that exceeds the number of molecules in the universe [López-Ortiz 1993]. The computational power needed to create the entire game tree of chess is far beyond what we could ever hope to achieved.

Since it is unrealistic to search the entire tree, computer chess concerns the examination of just a small sub-set of the game tree. From the knowledge gained by searching this sub-tree, a chess program attempts to choose the best move possible. Given this, there are two elements of the system that need to be determined:

1. How are the merits of the sub-tree terminals determined, if they are not game-theoretical terminals?
2. How will the sub-tree be traversed whilst collecting information from the terminal nodes in a meaningful fashion?

## 2.2 Static Board Evaluation Function

Given a terminal node of a chess sub-tree we require a means of converting the position, represented by the node, into a form that can be easily manipulated and understood by a computer. Since Claude Shannon's seminal paper [Shannon 1950] on computer chess this has mainly been accomplished by converting the position into a single numeral value using a *static board evaluation function*. The score returned by the static board evaluation function represents an estimate of the quality of the position for a side, generally from the viewpoint of the side to move.

The score returned by the evaluation function has the range [-∞, +∞]. If the position evaluated is a game-theoretical terminal then the score is easy to determine; if a WIN for the side to move then score = +∞, else if LOSS then score = -∞, else game is a DRAW and the score = 0. When the position being evaluated is a non-game-theoretical terminal then the static board evaluation function calculates an estimate of the quality of the board for the side to move. The function can only estimate the quality of the position as to calculate the *actual* quality of the position would require searching the position's game tree to its game theoretical terminals, a task that is impractical for non-trivial positions.

The quality of the position, for the side to move, is determined by taking the quality of the position for the opposition away from the quality of the position for the side to move. This property captures an essence of a zero-sum game, as chess is, in that no quality gain strengthens both sides simultaneously. Any quality gain is at the expense of the opposition, by the same quality amount.

$$\text{Evaluation}_{\text{(side to move)}} = \text{Quality}_{\text{(side to move)}} - \text{Quality}_{\text{(opposition)}}$$

The quality of a position for a side is calculated by examining the position for certain chess specific features. Chess players have long realised the importance of certain chess features, which are known to improve the chances of winning a game. They can be of either a defensive, offensive, positional or tactical nature and have been observed by humans during the long history of chess play.

Metrics or heuristics that evaluate these features are programmed into the evaluation function. When a feature in a position is recognised its associated payoff is added to the side's score. A payoff can be negative or positive depending on whether the feature increases or decreases, respectively, the side's chances of winning. The payoffs assigned to chess features have generally been extracted from human chess knowledge although there been work on deciding the values based on machine learning techniques [McConnell 1995]. Some of the most common computer chess heuristics are:

- Material – Material is probably the most important heuristic in an evaluation function as the number and type of a side's pieces have generally a strong correlation with a side's chances of winning. A standard weighting for the value of piece types is 1 for pawn (P), 3 for bishop (B), 3 for knight (N), 5 for rook (R) and 10 for queen (Q). The king has no material value since when it is captured the game is over.

- Pawn Structure – Pawns play a crucial role in any chess game and were famously described by the great 18th century chess player Francois-Andre Danican Philidor [Philidor URL] as "the soul of this game, they alone form the attack and defense". Passed, connected, isolated and doubled pawns are among the features examined for in pawn structure.

- Piece Mobility – The ability of a side to influence a game by its movement of pieces is critical to the game of chess. If a side's pieces have little mobility then the side's capability of attacking, supporting and evading other pieces is decreased, hampering the side's chances of winning. Mobility is especially important for bishop pieces.

- Piece-Square Values – The different squares of the board have varying importance and influence over the rest of the board. Squares in the centre of the board have greater importance and this is especially true if occupied by non-pawn pieces where the piece can exert their influence most. Squares at the edge of the board have lesser value as they are further from most of the action of the game and so cannot exert as much influence. The value given to the side that occupies a square is also dependent on the type of piece that occupies it. For example, a white rook at d1 would have greater value for white because of its attaching possibilities along the d file than a bishop, whose attaching possibilities would be limited.

## 2.3 Sequential Tree Search

### 2.3.1 Minimax Algorithm

Given the game tree of a zero-sum game the *minimax* algorithm [Levy and Newborn 1991] will return the move that will ensure the side to move its best possible result. The algorithm assumes that both sides will conform to *rational behaviour*, where each side is trying to maximise its payoff. The game of chess is a zero-sum game. Any gain by one side is at the expense of its opponent by the same payoff. A side seeking to maximise its payoff is simultaneously minimising its opponent's.

*Zero-Sum Game* $\Rightarrow$ $\text{Payoff}_{\text{Side 1}} + \text{Payoff}_{\text{Side 2}} = 0$

$\Rightarrow$ $\text{Payoff}_{\text{Side 1}} = - \text{Payoff}_{\text{Side 2}}$

$\Rightarrow$ $\text{Max}\ (\text{Payoff}_{\text{Side 1}}) = \text{Min}\ (\text{Payoff}_{\text{Side 2}})$

A small game tree is illustrated in Figure 2.3 with the terminal nodes evaluated from the perspective of the side to move at the root node. At the root node the side will choose the branch that will maximise the payoff. At the next level the other side will choose the branch that will minimise the payoff. At the next level the side will try to maximise the payoff and so on down through the levels. As side to move alternates through the levels so does the maximising and minimising of the payoffs. At the even levels the branches that lead to the successor nodes with the highest payoffs will be chosen and their payoffs adopted the parent node. At the odd levels the branches that lead to the successor nodes with the lowest payoffs will be chosen and their payoffs adopted by the parent node. In this fashion the evaluations from the terminal nodes are *backed-up* through the game tree to the root node.

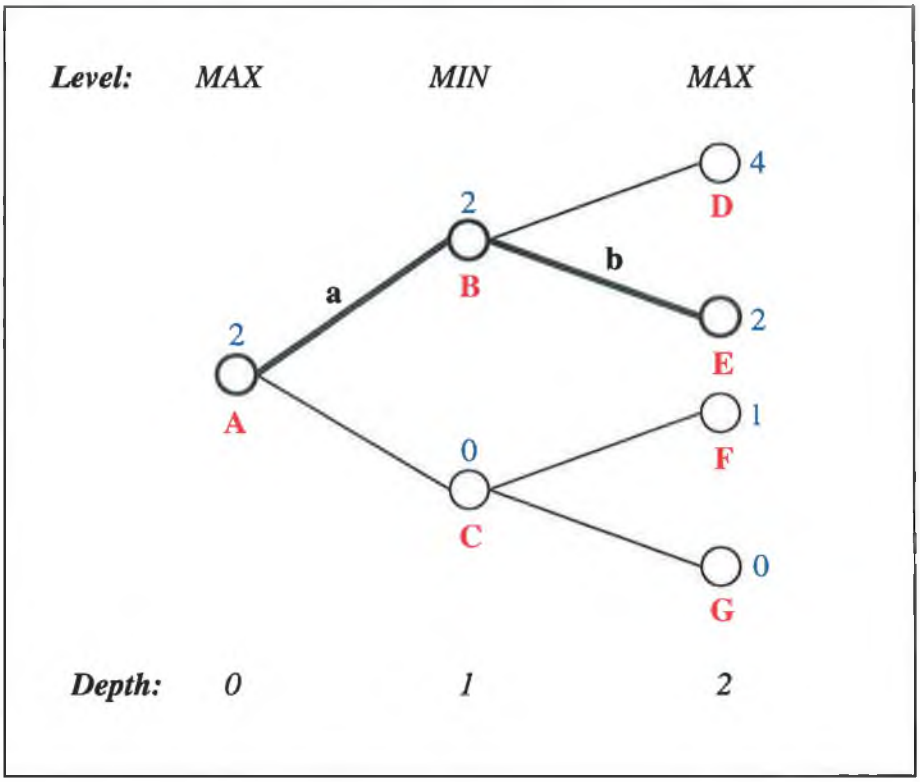



**Figure 2.3:** Game tree illustrating minimax algorithm.

At node B in Figure 2.3 the node has two successor nodes D and E with payoffs 4 and 2 respectively. Node B resides on a MIN level of the game tree, therefore the branch that leads to the successor node with the minimum payoff, node E, is chosen by the minimax algorithm. The payoff of node E is backed-up to node B so the payoff of node B is now 2. Node C, also on the MIN level, follows the same procedure and chooses node G and adopts the payoff of 0. All the payoffs of the nodes in level 1 have been determined and so we progress to the next level up the tree. Node A is in a MAX level and the payoffs of its successor nodes, B and C, are 2 and 0 respectively. The highest payoff between these is 2 and so the move that leads to node B is chosen. The series of moves that the minimax algorithm determines is best play by both sides is a-b and is referred to as the *principle continuation.*

Chess programs implement the minimax algorithm using a depth-first search strategy. A depth-first search strategy proceeds by resolving the deepest unresolved node. The flow of control in a depth-first minimax algorithm is illustrated in Figure 2.4.

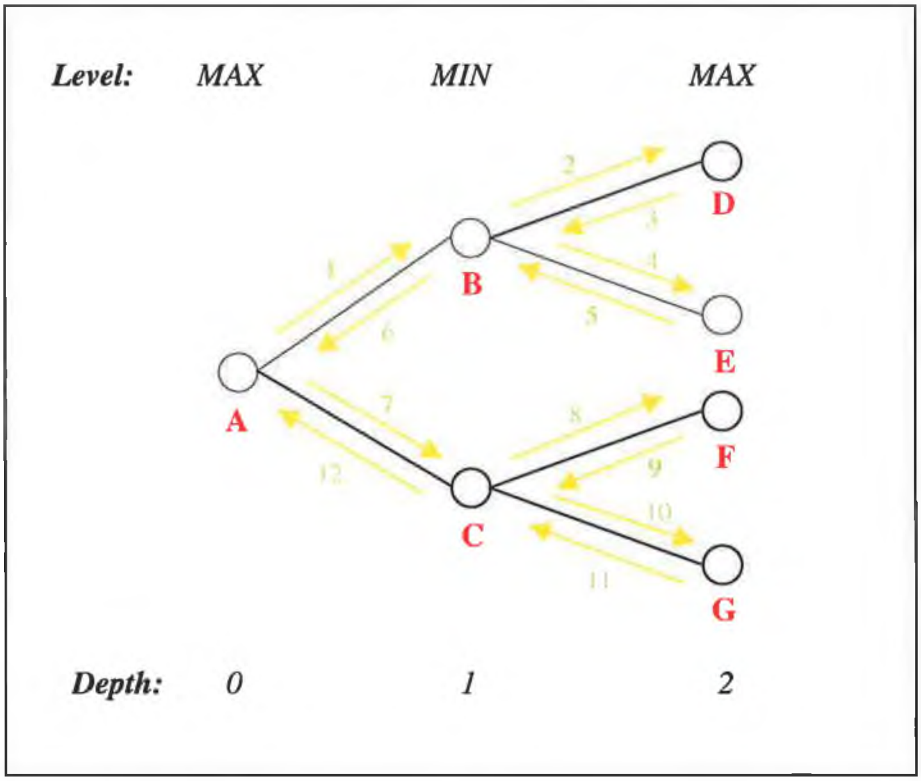



**Figure 2.4:** Depth-first control strategy of mini-max algorithm.

The depth-first search strategy has many desirable properties. The memory requirements of a depth-first algorithm grow linearly as it moves deeper into the game tree. This is of critical importance when searching deep into the game tree as alternatives can have memory requirements that grow exponentially, such as breath-first search. Flow of control is very simple in depth-first search, search can only move from a node to either a parent or child node of that node. This greatly simplifies verification of any program using the depth-first search strategy.

### 2.3.2 Alpha-Beta Algorithm

The alpha-beta algorithm [Knuth and Moore 1975] was a major improvement of the minimax algorithm. The alpha-beta algorithm allowed branches that will not improve the current back-up score to be safely ignored. Reducing the number of branches searched, and consequently the number of positions, allows the alpha-beta algorithm to search deeper into the game tree than the minimax algorithm in the same amount of time.

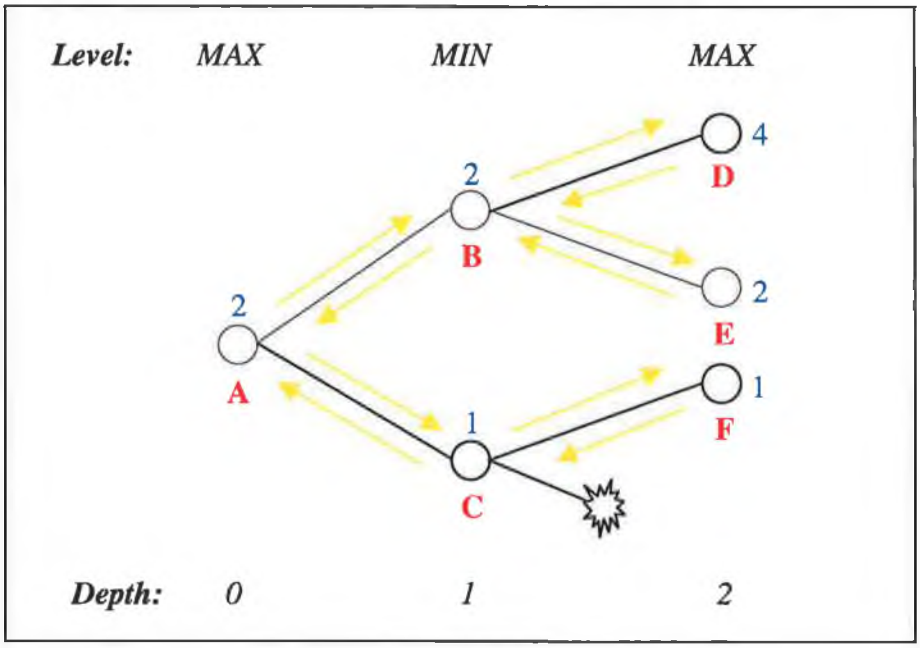



**Figure 2.5:** Game tree illustrating alpha-beta algorithm.

Figure 2.5 demonstrates the game tree in Figure 2.3 using the alpha-beta algorithm. After resolving node B and determining its value as 2, the least possible value of node A is 2. Node A is on a MAX level of the game tree and node B's value of 2 now needs to be exceeded if a node is to be preferred over B. The value of node F is 1 and therefore the maximum possible value of node C is 1, node C is on a MIN level and will not choose a node whose value is greater than this value. Node A can choose a move that leads to a value of 2 or a move that has at most a value of 1. Since node A is on a MAX level there is no need to resolve the value of node C's other successor nodes as node C will never be chosen over node B. An irrelevant branch to the outcome of the search has been ignored, called a *cut-off*, and only 3 of the 4 terminal nodes were determined. As the game trees get larger so do the size and frequency of the cut-offs.

The efficiency of the alpha-beta algorithm to achieve cut-offs is dependent upon the quality of move ordering at each node. If the alpha-beta algorithm searches successor nodes in worst to best order then no cut-offs can be achieved and its search tree will be the same size as the search tree of the minimax algorithm. For uniform trees of width W branches per node and a search depth of D ply, $W^D$ terminal nodes would be searched in this case. However, if move ordering is perfect with the best successor nodes being searched first then sizeable cut-offs can be achieved, reducing the number of terminal nodes of the searched tree to $W^{[D/2]} + W^{[D/2]} - 1$. This tree is referred to in computer chess as the minimal game tree.

### 2.3.3 Transposition Table

A transposition occurs when two or more different routes can reach the same position. Transpositions occur frequently in chess, as the moves available at a position are often also available for a number of plies further on so simple reordering of a side's moves can result in the same position being reached repeatedly. Transpositions occur frequently in human chess play, especially in situations where there are no immediate threats.

The transposition table works in the following manner. Before a position is searched the transposition table is first consulted to find if the same position had been searched previously. If the position is not found in the transposition table, the position is then searched as usual. After a position is searched, information such as its value and the depth to which it was searched are stored in the transposition table. Later if the same position is consulted for in the transposition table then the saved information in the table may enable the program to forgo searching the position again and instead substitute the saved information in the table into its calculations. The time saved by not repeatedly searching the same positions greatly outweighs the time spent consulting the transposition table and allows around an extra ply of the game tree to be searched [Verhelst URL].

### 2.3.4 Quiescence Search

Fixed depth game tree search suffers from being unable to consider the consequences of moves beyond the horizon of the search, this problem is called the *horizon effect*. A terminal position that appears good may lead to disaster just a couple of moves later. The accuracy of back-up scores depends on the quality of evaluations of terminal positions. Evaluations of dynamic positions, where pieces may be captured on the next move for example, do not give an accurate representation of the quality of the position. Positions that are less dynamic, often-called *stable positions*, are better evaluated by evaluation functions.

Quiescence search [Beal 1990] solves the problem of the horizon effect by performing further searches on terminal positions. Considering the full tree of these terminal positions is impractical because of time constraints. Instead quiescence search expands the more dynamic moves of a position, often these moves are restricted to capture, checks, check evasions and promotion moves. These positions are searched until stable positions are reached, which are then evaluated. Quiescence search provides a more accurate representation of the quality of terminal positions and hence the back-up scores of the game tree are of better and more accurate quality.

### 2.3.5 Iterative Deepening

Simple chess programs perform single deep searches of the game tree to a certain depth. In matches where each side has a set amount of time to move this method of play is unsuitable. A search may take longer than expected causing the program to choose a move prematurely before searching many of the branches. Alternatively, the search may terminate quickly wasting the extra time the side was given. Iterative deepening [Korf 1985] solves these problems by calling the fixed depth search routine many times with increasing depth values.

The series of searches finish when either the time limit or a maximum search depth is reached. Therefore a series of searches may begin with a 4-ply search, followed by a 5-ply search, followed by a 6-ply, and so on until either of the time or depth conditions is reached. The repeat searches of the same positions by iterative deepening may appear an inefficient and wasteful use of resources, but the benefits it yields greatly out-weigh these concerns. The exponential rate of growth of the chess game tree means that the sum of the searches previous to a deep search is small and almost insignificant compared to the deep search itself and can therefore be largely ignored. The shallower searches populate the transposition table with position evaluations and best moves. These entries may be used in later deep searches to improve move ordering and hence improve cut-offs in the game tree, quickening the time taken by the deep searches. In time controlled games iterative deepening is very efficient. If the time limit is reached and the current deep search is not complete the method can return the result of the previous shallow full tree search.

## 2.4 Problems with Current Computer Chess Techniques

### 2.4.1 Trade-off between Knowledge and Search

Theoretically, there exists two means of creating a perfect chess program. The first is a program that could search the entire game tree of chess to its game-theoretical terminals. For such a program, the game of chess would become a trivial task of backing up the results from the terminal nodes and selecting the move that ensured the best possible result. However, the astronomical size of the chess game tree means that such a program will remain an impossibility. The second method of perfect chess play is the implementation of a perfect evaluation function. Such an evaluation function could identify the result of a position from perfect chess play. Given a position the evaluation function could assign the proper win, loss or draw value for the board. The game of chess for such a program would become the simple matter of evaluating the successor positions of the current board and choosing the move that leads to the best result. A hindrance to the implementation of such an evaluation function is the problem of chess knowledge acquisition, a topic that will be discussed later.

Since neither search nor knowledge alone can feasibly enable perfect chess play; a combination of both must be used. As with any two competing sound ideas there exists a trade-off between them, which a program must properly balance. If a program implements too much chess knowledge the remaining computational power may not enable the program to search very deep, hindering the program's chances of success. Conversely, if the evaluation function does not implement enough knowledge then the terminal position evaluations do not adequately represent the positions to allow them to be compared properly causing sub-optimal solutions.

Although improving either the search or the evaluation component can increase the playing strength of a chess program; it has been the search component that has received the most attention from computer chess researchers. This has been due to the strong correlation that exists between the strength of a chess program and the depth to which it searches. Previous estimates suggest that an extra ply of search result in an increase of 100-250 ELO points [Feldmann 1997]. These increases, however, have been shown to diminish as one searches deeper into the game tree [Junghanns et al. 1997].

The strong relation between depth and strength meant that researchers concentrated most of their energies implementing and optimising their programs for larger and faster machines. Increasing computer chess strength lay more on software and hardware advances, which enabled deeper searches, than the arduous and difficult task of knowledge engineering for the evaluation function. The quick and easy benefits of deeper searches hindered the development of the evaluation function.

### 2.4.2 Problem of Knowledge Acquisition

The implementation of computer chess knowledge has been hindered not just by the attractive gains from deeper searches but also by the difficulties associated with extracting human chess knowledge. In order to properly implement human chess knowledge into an evaluation function, a description or system of human knowledge is required. Human knowledge relies on a process called intuition but how this process operates is unknown. It is believed by some that intuition is a human quality that cannot be implemented [de Groot 1965] while others believe that intuition is just a name given to rule based behaviour [Michie 1982]. The lack of a proper description of human intuition has hindered the direct integration of human chess knowledge into chess programs.

The presumption of intelligence to play chess makes chess a perfect vehicle for research by psychologists. As a consequence of understanding how humans play chess they gain a better understanding of intelligence in general. The work of psychologists into chess has exposed some of the problems of transferring human chess knowledge into computer programs.

From the pioneering work by the Dutch psychologist Adriaan de Groot [de Groot 1965] it is known that human chess play relies far more on chess knowledge than deep searches of the game tree. De Groot found that the number and depth of branches searched by a grandmaster was indistinguishable from that of an ordinary club player. Both players examined seldom more than 100 different branches, even in difficult positions. The only difference between the two players' searches was that the grandmaster searched the more relevant branches and better moves. Before becoming a grandmaster he/she will spend many years in chess training. Over this time the grandmaster acquires a vast quantity of chess knowledge that enable him/her to instantly recognise the better moves available at a position. The better moves appear to the grandmaster intuitively while the weaker moves are ignored. Because of a grandmaster's ability to find the better moves, he/she can perform deeper searches on just the better lines of play as opposed to the full tree search method used by modern computer chess. The details of how a grandmaster recognises these better moves are required if computers are to play chess using the same method as humans.

When a chess grandmaster views a chessboard in the course of a game he/she groups the pieces together into different familiar patterns called *chunks* [Chase and Simon 1973]. These chunks have been previously encountered and the player instantly recalls the opportunities and dangers associated with them. This information is then used to direct the player's search to the most relevant lines of play. The chunks and their information are built up over many years of chess training. The number of chunks and its information is estimated to be a minimum of 50,000, this number is "comparable to the typical language vocabularies of college-educated persons" [Simon and Schaeffer 1992]. Most of this information works on a subconscious level during chess play. This subconscious information plays a part in a grandmaster's feeling of intuition in chess play but the process of how this information is collected, organised, accessed and evaluated is unknown. Because this information is mostly held on a subconscious level it is difficult for a grandmaster to explain all the reasons for their choice of moves.

The chess knowledge that a grandmaster describes is often vague, interrelated and incomplete and difficult to convert into simple rules that could be implemented into a chess program.

Work has been performed on trying to incorporate the idea of chunks into move generation in computer chess [Sinclair 1998] [Flinter and Keane 1995]. This work involved the automatic extraction and storage of chunks from positions by chess masters along with the move chosen and the result of the game. By searching a database of chunks from positions that occurred in thousands of games, a ranked list of moves to search first is created. The lists of moves were hoped to create good cut-offs and hence allow deeper searches of the game tree. Chunking has a number of computational drawbacks, however, that make it currently impractical to use. The discovery and classification of chunks in a position is computationally complex and the delay involved when querying the database of chunks both make chunking impractical for current chess programs.

Despite the lack of chess knowledge, computer chess has made incredible progress, culminating in Deep Blue's famous victory [Deep Blue URL] over Kasparov, then World Champion, in 1997. This achievement was the direct result of advances in chess search; Deep Blue was capable of evaluating and searching 100-200 billion moves within three minutes. The evaluation function, however, returns only an estimate of the value of positions and therefore contains error. As modern computers reach their limit of search capability this error will need to be reduced, by increasing chess knowledge, if further progress is to be made in computer chess.

# Chapter 3

# Kasparov versus The World

The purpose of this chapter is to introduce the idea of collaboration in the game of chess and to give an example of it in the real world. In order to demonstrate chess collaboration in action, the contest 'Garry Kasparov versus the World' is reviewed, detailing the rules of the contest and the contest itself. The research presented in the rest of this thesis does not take the Kasparov versus the World contest as a model of collaboration to be recreated in computer software but rather uses the contest as an inspiration for the use of collaboration in computer chess and to pose the hypothesis that collaborating shallower searches are as good or better than a single deeper search. A direct model of human chess collaboration is beyond what is currently possible, as will be discussed in Chapter 4 of this work.

## 3.1 Background of Contest

On June 21st 1999 Garry Kasparov made the opening move in his historic chess game against the 'rest of the world'. The contest, which was organised by Microsoft Corporation and which was hosted on their MSN network of Internet services, gave players from around the world the opportunity to participate in a game of chess against the then world chess champion. The event attracted over 3 million people from 79 countries, making it the largest online event at the time.

The motivation behind Microsoft's involvement in the game was simple. IBM's Deep Blue (1996) and Deeper Blue (1997) versus Kasparov matches [Deep Blue URL] were a public relations bonanza for the company, receiving positive international coverage. Microsoft hoped their amalgamation of chess and computer competition would also achieve this level of success. The Deep Blue matches had showed the power of computation, the Microsoft contest hoped to show the power of the community.

### 3.2 Details and Rules of Contest

The contest between Kasparov and the World consisted of just a single game of chess with Kasparov playing as white. Obviously a single game does not provide a proper examination of the sides' playing strengths but time constraints prevented more than just the single game being played. Each day side to move alternated between the sides, giving each side 24 hours to make each of their moves. When concluded the contest's single game took over 4 months to play. The contest was organised more for entertainment than science and so only the single game was ever organised.

"The World" was to be guided by four World Team Coaches who were all teenage chess experts. They were [Microsoft PressPass URL]:

- Etienne Bacrot, 16, became the youngest grandmaster in history at only 14 years old.

- Florin Felecan, 19, was the highest-rated American chess player under 21.
- Irina Krush, 15, was the U.S. women's chess champion and youngest member of the U.S. Olympiad team.
- Elisabeth Pähtz, 14, was ranked eighth in the World Championship of youngsters and was a member of the female German National Chess Team.

The World team had two main means of communicating together. The first was the official website of the contest. The website displayed the recommended moves by the World team coaches and allowed players to vote for the next move to be played by the World team. The second means of communication for the World team was the World Chess Strategy bulletin board. The bulletin board allowed players and coaches to discuss tactics and strategy for the game and the advantages/disadvantages of the different moves available to them. To preserve the integrity of the game Kasparov agreed not to visit the website or the bulletin board during the contest thus allowing the World team to freely discuss their tactics and strategy.

The process that the World team initiated for each of their moves was as follows. Kasparov's previous move would be posted on the contest's official website with the new current board position. The World team then had 24 hours to decide on their next move. Coaches and players would independently analysis the position. While analysing the position coaches and players could communicate to each other using the bulletin board. This allowed coaches and players to trade ideas on the different options available thus allowing them to influence each other's opinion of the game. When satisfied the coaches then nominated the move that they thought the World team should play next. These were then displayed on the official website. Players from around the world could then vote for the nominated move that they preferred most. Each players' vote had equal weight, meaning that a vote cast by a chess beginner had just as much influence as that of a master's. When voting was concluded the nominated move with the highest vote was played by the World team side. Kasparov then had 24 hours to make his move after which turn to more returned to the World team and the process began again.

The game of chess requires strategy and tactics, which involve sequences of moves being played. It may appear that the multi-player/voting system is inherently flawed as the possibility exists that move sequences could be cut-off halfway through. These cut-offs could be the result of a number of different camps of players, with different opinions on strategy and tactics, winning the vote for their recommended move at different times. However, this possible chaotic play could work as an advantage to the multi-player team. It can be assumed that the coaches and players are rational players, meaning that they would only recommend/vote for moves that they feel are best. When recommending/voting they will be aware of the different move sequences available to them. There is no reason for any of them to cut-off a move sequence unless they feel that there is a better move available. Thus move sequences should only be cut-off if a majority of players feel that an advantage to the World team is to be gained by cutting them off. This is perfectly good chess play. This could be an advantage for the multi-player team as it makes it less predicable and harder for its opponent to anticipate.

### 3.3 The Match

The game began by Kasparov making his opening move on the June 21st 1999. The opening sequence followed a standard variation of the Sicilian defence until the World's 10th move. The World, playing as black, broke from traditional openings with an unusual queen move that led to an exchange of pieces and a complicated middle game position. Kasparov described the move as "a brand new idea" that put the "pressure on black".

As the game progressed it became apparent that the high quality of play by the World was in a large part due to the tremendous effort of one of the World team coaches, Irina Krush. She became the unofficial World team leader and her analysis of the game and quality of work outstripped that of all the other coaches Her regular participation on the World Team Strategy bulletin board made her a favourite among the thousands of players throughout the world. Krush's moves were a synthesis of her own ideas, several grandmasters' and the ideas shared on the World Team Strategy bulletin board from regular chess players. Of the initial 57 moves made by the World team, 53 of them were Krush's recommended moves (the exceptions being moves 3, 6, 51 and 52).

It may appear that Krush had undue influence over the game and that the game could be described as merely a game between Krush and Kasparov. This view, however, would ignore the strong influence that the other World players had on the game. First of all, the World team did not always follow the moves recommended by Krush. Each move has the potential to transform and influence the rest of the game and cannot be ignored when assessing the influence of the World players on the game. The view also fails to appreciate the influence that the World players had on Krush. Krush regularly participated on the World Chess Strategy bulletin board where the analysis and suggestions from other World team players would have strongly influenced her opinions of the game and her choice of move. Her recommended moves were not from her sole analysis of the game but from a synthesis of many people's analysis of the game combined with her own. If Krush had been played the game on her own, it is very unlikely that she would have played the same moves. Krush at the time of the contest had an ELO rating of 2375 and Kasparov had a rating of 2849. When the difference between ratings is more than 400 then the higher rated player is expected to win 100% of the time. The Kasparov versus the World match was much closer than this and suggests the World team was playing better than Krush could have alone. Therefore I believe it unfair to reduce the Kasparov versus the World game as merely Kasparov versus Krush. However, the model of collaboration employed by Krush, using other players' analysis to supplement her own, does appear to succeed and could be used as the basis for collaboration in computer chess.

The game proceeded as planned and was an incredible online success, however, from move 51 on things started to go wrong. It was black's 51$^{st}$ move and most of the bulletin board community had decided that 51… Kb1-a1 was the best move for the World team to make. Irina Krush and the Grandmaster Chess School also recommended this move. However, when voting was complete a weaker move suggested by Elisabeth Pähtz, 51… b7-b5, received the most number of votes and was therefore played. Microsoft had previously denied that "vote-stuffing", illegal multiple votes by the same player, was possible. Yet following black's controversial 51st move a bulletin board member, Jose Unodoes, claimed that he had "stuffed the vote" by simple entering multiple email addresses. Microsoft responded to this and other "vote-stuffing" claims by only allowing players using Microsoft Windows to vote. Players who had been using non-Windows OS platforms from the beginning of the game could now no longer participate in the contest. This was a cause of great concern and protest but it was only the beginning of the problems for the contest and Microsoft.

Due to email problems Krush received Kasparov's 58th move later than expected and subsequently her move recommendation was expected to be delayed. The World Team Strategy bulletin board had determined that 58… Qf3-e4 was an expected loss and that 58… Qf3-f5 gave the best chances of achieving a draw. Two of the World Team Coaches, Pähtz and Bacrot, did not participate in the bulletin board and recommended 58… Qf3-e5, the weaker move. The other coach, Felecan, recommended the stronger move 58… Qf3-f5. The moderator of the game, grandmaster Danny King called 58… Qf3-e4, the weaker move, a "sensible option". World Team voters that did not follow the bulletin board only saw that the weaker move was recommended by a 2:1 majority of the team coaches and that the move had been described in good terms by the game's moderator.

Not wanting to delay voting for Krush's recommendation Microsoft began the voting process. The website indicated that "Irina's [Krush] move recommendation would appear here shortly". What happened next is unclear. Krush emailed her recommendation to Microsoft but the contest's website was not updated to reflect Krush's recommendation. Had Krush's move recommendation, which was for the stronger 58… Qf3-f5 move, been published on the website it was very likely that it would have been accepted as the move to play by the World, the World had previously played most Krush's recommended moves. However, without the benefit of Krush's analysis The World voted for the move that the bulletin board and Krush deemed to be a loss, 58… Qf3-e4. Maybe as a protest to Microsoft or because she felt the game was over anyway Krush resigned as a World Team Coach while the game was still running.

Following the tactical error of move 58 and the resignation of Krush, many of the World players felt the game was doomed. As protest to the incompetence of Microsoft's handling of the affair, the players on the World Chess Strategy bulletin board organised mass voting for 59… Qe1, a move that was suicidal and gave away a queen for free with check. When voting was concluded 66.27% of the votes were for the suicidal move. Anticipating the public embarrassment of the World team concluding the game in this manner, Microsoft disqualified all the votes for the move claiming allegations of "vote-stuffing". This decision, by Microsoft, caused a furore of anger and public protest.

The game concluded on October 23rd 1999 with the World team resigning in the face of a winning position for Kasparov. What had progressed well for the majority of the game had ended in a storm of controversy.

## 3.4 Summary

Despite the controversy that marred the end of the Kasparov versus the World contest, it was an online success story. The contest became one of the "largest gaming events in history" [MSN Kasparov URL]. The game itself was far more exciting than anyone anticipated. A sentiment Kasparov endorsed by describing it as "the greatest game in the history of chess".

Daniel King, the moderator of the contest, described the game as "a genius versus a committee" at its beginning. The quality of chess produced by the 'committee' was one of the great surprises of the contest. The 'committee', comprising of the four expert World Team Coaches, lead by Irina Krush, and the World Chess Strategy bulletin board discussions from players around the world, produced moves of play that challenged the chess strength of Kasparov. Whether the World team could have beaten Kasparov if the mistakes of moves 51 and 58 had not occurred will never be known.

During the contest both sides were allowed the use of computer chess programs, a sensible decision, as banning their use would have been impossible to enforce. However, this questions whether the playing strength of the World team could have been the result of the chess programs instead of the team approach. Without doubt the playing strength of many of the World team players were enhanced by the use of chess programs. This would have slightly increased the playing strength of the team as players who used chess programs may have provided better analysis of moves that coaches and other players may have incorporated into their own analysis of the game and hence improving the playing strength of the team. However, the influence of computer chess programs on the playing strength of the World team should not be over-blown. Strong national masters will regularly beat the vast majority of chess programs, particularly the publicly available chess programs that World players would have used. International masters and grandmasters would be expected to beat chess programs 100% of the time under normal playing conditions. Even under blitz (all moves in 5 minutes), which suits chess programs more than humans, grandmasters consistently outperform the chess programs. The World versus the World contest was at the opposite side of the time spectrum with each move taking 24 hours. At the level of grandmaster chess play with 24 hours to make each move any advantage of using chess programs diminishes to almost zero. Therefore, the playing strength of the World team should be credited to the team's collaboration rather than any use of computer chess programs.

# Chapter 4

# Collaborative Computer Personalities in Computer Chess

As has been discussed in Chapter 2, it is impractical for computer chess programs to search the entire game tree of chess. Instead they search just a small sub-tree of the tree and use an evaluation function to estimate the values of the positions at the sub-tree terminals. The evaluation function assigns a value to a position by searching the position for positive and negative board features that could influence the proceeding game. The board features are assigned different weightings depending on their estimated importance. By changing the weightings of the board features different importance can be imparted on to them resulting in the chess program playing with a different style of play.

For example, we will take an evaluation function from a chess program whose defensive board features have been heavily weighted. This defensive evaluation function will more highly rate defensive positions, which contain these heavily weighted defensive board features, than positions that do not contain the defensive features. The purpose of the search algorithm of a chess program is to decide on the next best more based on the choices of moves that each of the sides have. The search algorithm will choose the moves with the best payoffs for each of the sides. The positions with the highest payoffs will be those that were highly rated by the evaluation function, the positions with strong defensive features. Therefore the search algorithm will view the game tree from the perspective that both sides are playing for defensive positions. If a human tries to play for defensive positions they can be described as having a defensive style of play and so it is also with chess programs that have defensive evaluation functions. This correspondence between evaluation function and style of play extends to many different styles of play by weighting the board features in the evaluation function differently.

Modern computer chess is based on the paradigm that there is one evaluation function, or style of play, for a given position that is superior to all others. Each terminal position is evaluated from the perspective that only one style of play will follow it. This is different from the idea of only one evaluation function being present in a chess program. Programs may implement different board feature weighting schemes for different stages of a game or in the case of the M* algorithm [Carmel and Markovitch 1994] there may be separate evaluation functions for each side. In these cases, however, a side only evaluates a position using one evaluation function, or style of play for any given position.

The Kasparov versus the World contest suggested another possible model of play for computer chess. In the contest multiple players analysed the same positions each using their own individual style of play. From the analysis, four moves were recommended to be played next and the players in the team voted on which of these they preferred. The recommended move that was deemed best from the voting was then played. The Kasparov versus the World serves only as an example of a possible collaboration mechanism and will not be directly transferred to software for this work. The World team played exceptionally well and may have benefited from its ability to view and play the game from many different styles of play. This model of play could possibly be adapted for computer chess, allowing a program to view the same game tree from multiple styles of play and achieving better chess play.

## 4.1 Possible Advantages of Collaboration

At different positions in a game it may be better to play with a different style of play. For certain positions it may be better to play aggressively for example. How to decide what style of play should be followed is very difficult to formalise and chess masters themselves would disagree on how many positions should be played. The choice of move and the style of play to be used very much depend on the personality of the player and their feelings on the game.

Modern chess programs view each terminal position from only one style of play. They do not consider the same position from the various styles of play that could follow. Because of this, tree branches that may lead to better positions may be ignored because they require to be appreciated from a different style of play than that used by the chess program. This may result in sub-optimum play. Searching the same game tree with multiple collaborative styles of play may enable the chess program to overcome this problem. It may be possible for the program to choose the tree branches that lead to better positions by viewing the terminal positions from the perspective of the style of play that should follow. The program would be able to choose between the various styles of play available during the game depending on the position.

If an opponent can accurately anticipate one's following moves then he/she has the ability to dictate the game and steer the game towards game states that are advantageous for him/her. The ability of the multiple style of play program to choose between different styles of play may make the collaborating team much harder to anticipate and hence more difficult to beat.

## 4.2 Possible Disadvantages of Collaboration

During runtime a chess program has a set amount of resources to search and evaluate the game tree. A program that has multiple styles of play may not be able to devote as many resources to each style of play than a program with only one style of play. This decrease in resource allocation may result in the multiple styles of play not being able to search as deeply into the game tree as the single style of play program. Depth of search has been shown to be a major factor in a program's playing strength. Therefore the possible decrease in search depth may be a possible disadvantage of using multiple styles of play in a chess program.

Another possible disadvantage of collaboration in computer chess is that the style of play that may be considered to be in the middle ground between the other styles of play may get too much influence on the outcome of the game. For example, if in a chess program that used an aggressive style of play, a defensive style of play and a style of play that was between an aggressive and a defensive style of play then the latter style of play could possibly dictate the play for most of the game. The reason for this may be that the aggressive and defensive styles of play are so different from each other that each would rather the recommended move of the middle style of play than each other's. The middle style of play would become the most dominant style of play and the team would accept most of its recommendations. This would result in the multiple style of play program effectively becoming just a single style of play program with the added disadvantage of decreased resources allocation as discussed above.

### 4.3 Constructing Chess Personalities

For the remainder of this work a *personality* will refer to an entity with a certain style of chess play. So Kasparov who has an aggressive style of chess play can be described as an aggressive personality. While Karpov who has a positional style of play can be described as a positional personality [Chess Corner URL]. In human chess, style of play would be associated with features of the board having different importance to each other. For example, an aggressive player would care less about king safety than a defensive player would. When analysing a position a human player would try to steer the game towards positions that contained the features deemed desirable by their style of play.

In computer chess the idea of personality is located within the evaluation function. The evaluation function analyses positions for features that it feels will influence the outcome of the game that follows and assigns a score to positions based on the payoffs assigned to those features. So, if an evaluation function assigns more importance to pieces in close proximity to the opponent's king then that evaluation function combined with a search function would have an aggressive personality.

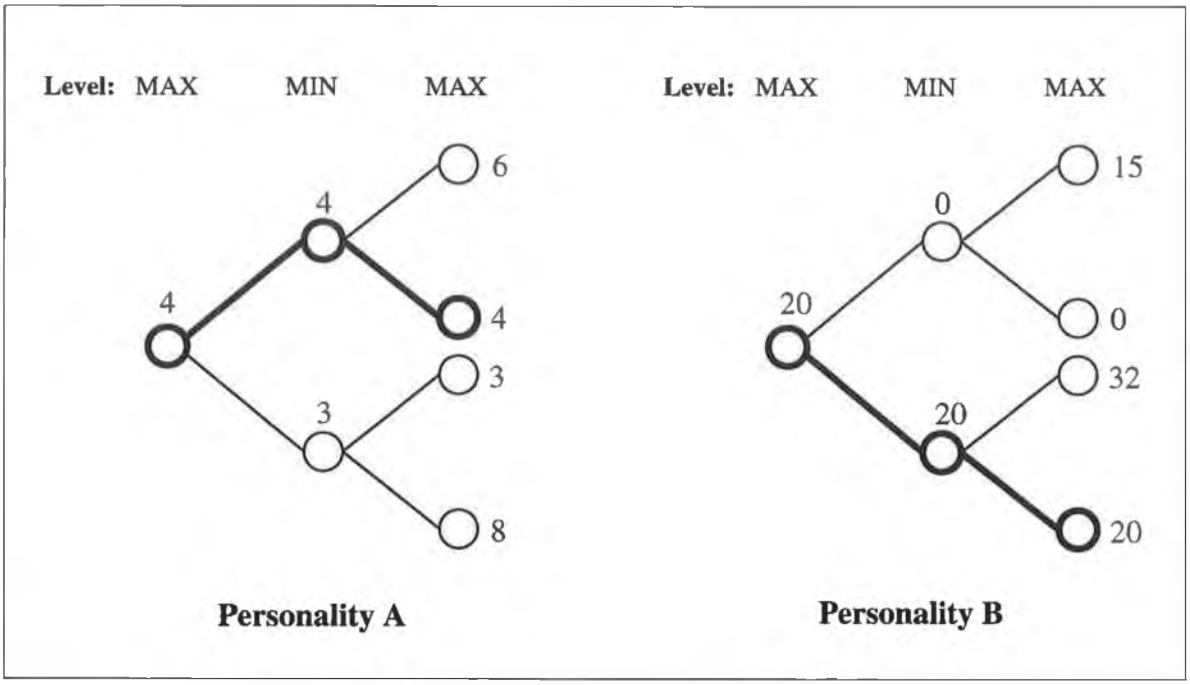



**Figure 4.1:** Diagram illustrating the same game tree from the perspective of two different personalities using the minimax algorithm.

Figure 4.1 demonstrate the same hypothetical game tree viewed from the perspective of two different personalities. The evaluation functions of each of the personalities assign values to the terminal nodes of the game trees. Because different evaluation functions assign different payoffs to board features the values returned by the functions for the same position could be different. Using the minimax algorithm the principle variations (PV) of the trees are calculated and are indicated in bold. In the case of the diagrams, different PVs have been selected for each of the trees. Neither of these PVs is incorrect, both are valid answers for the different evaluation functions used. If the different personalities in Figure 4.1 are to co-operate then a method of combining and resolving their PVs and evaluations is required.

Crucial to this work is that while the personalities collaborate they still maintain their own individual style of play and that the moves played by the collaborating team are recommended by at least one style of play. The most obvious approach to combining and resolving the personalities views of the game tree is to normalise and average their evaluations of the game tree terminals. However, this method would eliminate the team's ability to play the game with the personalities different styles of play. Effectively this method would take the personalities and average them to create just a single personality with a single of play. A much simpler method of getting the same result would be to just have a single evaluation function whose weightings are an average of those used by the multiple personalities. The ability for the team to play with different styles of play at different times would be eliminated.

## 4.4 Designing Multiple Personalities Solution Methods (MPSMs)

If multiple personalities are to collaborate a method of combining/analysing their searches is required. This task will be the responsibility of various Multiple Personalities Solution Methods (MPSMs), the details of which will be discussed in Chapter 6 of this work.

In this work all the personalities in a collaborating team will be deemed to have equal chess strength. This means that none of the recommended moves by the personalities will be given automatic preference over the others and that the voting power of the personalities will be equal. This may seem a hindrance for the collaborating team, as a personality with a seemingly weaker style of play will have as much influence over the game as that of a stronger player. However, this is not as serious as it may appear. If for a given position a certain personality is particularly weak and its choice of move is unwise then the other personalities should recognise this and the team will not play its choice of move. This personality should still retain the same influence, however, as at other positions its style of play may be perfect and any reduction of its influence may hinder the team from choosing its recommended move. The choice of which of the personalities recommended moves should be played by the collaborating team will be decided upon by taking into account the preferences of the personalities of the different recommended moves. MPSMs will define sets of rules that will measure the personalities preferences of the recommended moves and the move the collaborating team will play will be decided upon by this.

### 4.4.1 Limitations Imposed upon MPSMs

*Cannot use human collaboration as model*

Initial ideas for a MPSM considered human chess collaboration as its inspiration. When investigated further these ideas soon lose their practicality. Human collaboration depends not only on domain knowledge but also on the environment in which the collaboration takes place and the social dynamics that develop in that environment. Modern computer chess techniques are not designed/suitable for multiple searches interacting together. They are designed for independent searches of the game tree, expanding branches in a systematic brute force fashion. Human collaboration is much more interactive than this with players being able to interact at each move and their interaction influencing each other's game tree. Therefore modern computer chess techniques are unsuitable for a chess program that uses human collaboration as its model. Also, any method of human chess collaboration will need to use human chess knowledge as its foundation. As has been discussed in Section 2.4, implementations of human chess knowledge have proved beyond our reach, making any use of human chess collaboration techniques impractical. Therefore, the collaboration method by computers will be radically different from that of humans.

*Limitations of Data returned by Minimax Algorithms*

The capabilities of MPSMs are dependant upon the amount of data that they have access to. Most chess programs use enhanced versions of the minimax algorithm, such as alpha-beta or PVS. These search algorithms output only two major pieces of data; the PV and the backup score of the PV's terminal. The capabilities of any MPSM will be limited by this scarcity of data for each search performed.

In order to use this data, a means of comparing the results of two searches with different personalities is required. PVs cannot be directly compared. PVs are just series of moves; they contain no measurement of their goodness. The scores from searches are measurements and are open for comparison. By comparing scores, measurements of PVs can then be inferred. This is, however, not as straightforward as it appears.

Utility theory is a branch of mathematical game theory, developed by von Neumann and Morgenstern [von Neumann and Morgenstern 1944], that is concerned with the values assigned to outcomes in games. In utility theory the utility function is "a quantification of a person's preferences with respect to certain objects" [Davis 1997] and represents the evaluation function of computer chess.

One of the major results of utility theory is that utility scores from different functions are not comparable [Luce and Raiffa 1957]. For multiple personalities trying to collaborate this adds the restriction that although a personality's scores are comparable, comparisons of different personalities' scores are not allowed.

| Personality | PV | Score |
|---|---|---|
| A | a-b | 3 |
| A | c-d | 5 |
| B | e-f | 4 |

**Table 4.1:** PVs and scores from personalities A and B.

To illustrate this, in Table 4.1, a list of different PVs and scores by personalities A and B are given. Utility theory allows the deduction that the series of moves a-b by A, with a score of 3, is inferior to the moves c-d, with a score of 5. Deducting that B's moves of e-f, with a score of 4, is inferior to A's c-d moves, however, is not possible as the scores are from different utility functions.

### 4.5 Summary

The Kasparov versus the World match suggested the idea of multiple personalities collaborating to create better chess play. The task of combining/analysing the PVs and scores from various personalities will be performed by a MPSM. This task is hindered by the lack of knowledge of how humans perform the task, the limited information that standard search algorithms return and by the constraints imposed by Utility theory. By making the MPSMs more sophisticated it is hoped that the quality of chess play by the multiple personalities will improve.

# Chapter 5

# Test System

The development of a computer chess program is a very complex software-engineering task. Computer chess encompasses many game specific concepts that are not present in other software engineering tasks. This impedes general understanding of the concepts required for computer chess and therefore complicates the creation of computer chess programs. Speed is a very important factor to the success of a computer chess program. The program code needs to be fast and highly optimised; a task complicated by the presence of interrelated and complex computer chess concepts. Computer chess programs generate an enormous amount of runtime information. The chess program DarkThought visits 200,000 nodes per second on a 500MHz DEC Alpha system [Heinz 1997]. Each node visited would have runtime information associated with it such as alpha, beta and depth values. This amount of information allows errors to easily go unnoticed and complicates the debugging process. To create a computer chess test system one needs to overcome these issues and more.

## 5.1 Process Control Used by Test System

The system requires a means of controlling the amount of execution power that the multiple personalities collaborating may use. If the multiple personalities were given too much or too little power, this would skewer the results. A means is needed to control the power they are given so that meaningful results may be extracted.

The standard method for control in tournament chess programs is time, the same as standard human tournaments. However, for this research I propose that depth control would be superior for the following reasons:

1) Results are easily re-createable using depth control. Repeated depth controlled searches are only dependant upon simple input data. Time control depends upon the amount of CPU given during the period of search; other programs running, system interruptions and changes in hardware can affect this.

2) Complexity of time control. The amount of time given to a move is a complex issue in computer chess. Standard games assign a set amount of time available to the sides for all of their moves. If they exceed this time constraint, they automatically lose. Implementing such a control for multiple simultaneous searches would be a complex task and would be hard to know if it hindered the progress of the multiple personalities.

3) If time control is used each of the personalities may have reached different depths into the search tree when time is up. Each of these results would not have equal importance, the search that returned the deepest PV would be more likely to be the superior to the rest. Depth control would not have this problem, as all the PVs would be of equal depth.

## 5.2 Distribution of Processes

In order for computer personalities to collaborate each of the personalities may perform several searches of various positions. Each of the personalities searches are independent of each other, they do not share any dynamic information that is required while searching. Therefore to decrease the execution time of the test system, many of the searches can be distributed among different machines.

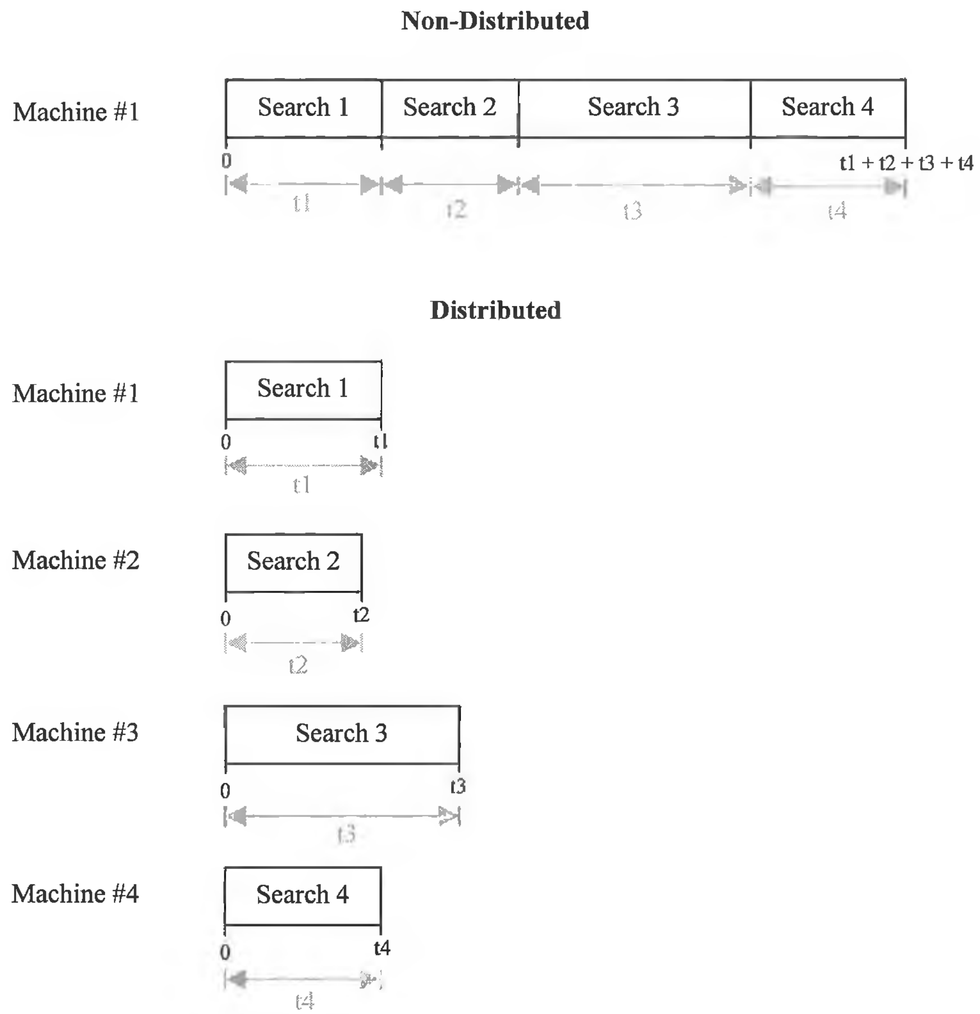



**Figure 5.1:** Time taken for multiple searches distributed and non-distributed.

This reduction in execution time is illustrated in Figure 5.1. Searches do not all take the same amount of execution time. Distributing the processing reduces the time taken from the sum of the search times to the time taken by the longest search. These savings greatly improve system-testing times.

There are various methods of implementing the distribution of searches such as CORBA [CORBA URL] or a client/server architecture [Orfali et al. 1999]. However, for the test system the IBM Aglets framework [Lange and Oshima 1998] was used.

### 5.2.1 Aglets Overview

The Aglet framework is an API and a set of interfaces to create mobile Java agents. Aglets have the capabilities of transporting themselves, that is their program code and current data, from one computer to another computer. The Aglet framework was created at the IBM Tokyo Research Laboratory [IBM Tokyo URL] in Japan in 1995 and was released under the Open Source Initiative in 2000.

Before discussing the design of the test system using the Aglet framework, a short description of agents and details of the Aglet framework will be given.

*Aglets as Mobile Agents*

The term 'agent' has become one of the buzzwords of software engineering for the last number of years. Each year new products and services are released promoting their use of agent technology when in effect they contain little that is new or different from a pre-agent perspective. The reason for this is that so many people have defined what their term of 'agent' means that the term itself has come to mean almost any piece of software [Franklin and Graesser 1996].

The Aglets project views software agents as "programs that assist people and act on their behalf. Agents function by allowing people to delegate work to them" [Lange and Oshima 1998]. This definition is extended by Aglets to create mobile agents that have the ability to transport themselves from one computer to another in a network. When an aglet moves it transports its program code and state information, which allows it to continue its execution from its state prior to transportation. The Aglet API provides aglets with the abilities to travel from host to host, to create and destroy themselves and other aglets, and to communicate with other aglets.

### 5.2.2 Design of Aglet System

The crucial component of agent technologies is their ability to delegate tasks to an agent. For an elegant design, tasks need to be delegated in a meaningful and intuitive fashion. An agent should be assigned a particular role and delegated only tasks associated with that role. The agent should not be delegated tasks outside this role as this would obscure and unnecessary complicate the design. Just as a class in object-oriented design should only implement functions associated with the class type, so should an agent only be delegated tasks associated with its role.

The aglet test system comprises of three different agents; each with a different role. The roles were assigned based on the roles required by the game of chess. The three agents are:

1. Referee agent, which manages and umpires series of games.
2. Team agent, which represents a side in a game of chess. A side may comprise of one or more player agents.
3. Player agent, which represents a chess playing personality.

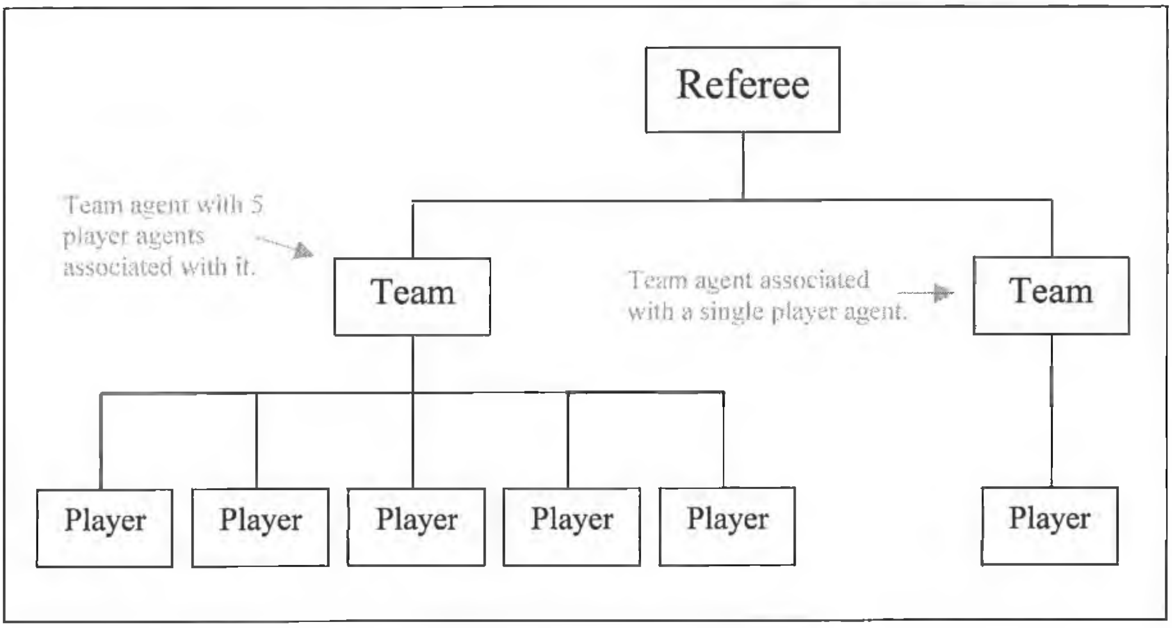



**Figure 5.2:** A possible configuration of agents in a series of games

For a game, a referee agent will have two team agents associated with it. Each of the teams represents a side in a game of chess (i.e. black or white). Each team has associated with it one or more player agents. Both teams are associated with a referee agent and each player agent is associated with a team agent.

The purpose of using agent technology is to allow the distribution of processing. To achieve this each of the agents during runtime is assigned to a specific machine. When an agent is created its intended location is passed to it as a parameter. The first task performed by an agent is to transport itself to its intended location. Once there, the agent then sends a message to its parent indicating that it has reached its destination.

*Referee Agent*

The role of a referee agent is to manage multiple consecutive games of chess. The referee agent is responsible for the creation of the games' teams and the interaction between the game and the teams. The referee passes the current game board between the teams and handles their chess moves.

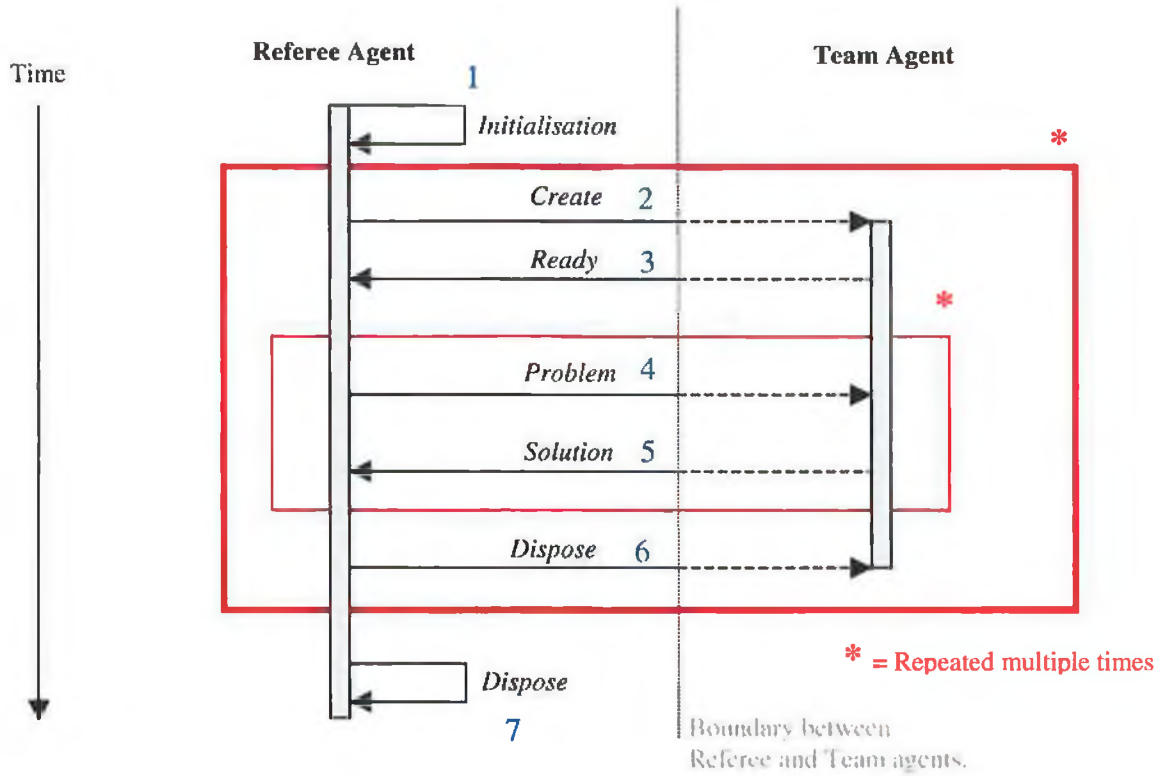



**Figure 5.3:** Time flow diagram of Referee agent's interactions.

The following steps detail the actions of the referee agent as indicated in Figure 5.3:

1) The referee agent is created. The referee will manage multiple games and begins with the first of the series.
2) For the current game, create the two team agents.
3) Wait for the two team agents to respond that they are ready.
4) Send current game board to team agent of side to move.
5) Receive solution from team agent and update board. If game is over then go to step 6 else return to step 4.
6) Dispose of both team agents. If no more games in series of games then go to step 7 else set to next game in series and return to step 2.
7) Dispose of referee agent.

*Team Agent*

The role of the team agent is to manage the players associated with the team and to communicate with the referee agent. The team agent does not search the chess game tree itself but chooses the best move to make from the searches performed by its player agents. The team agent creates player agents that have been assigned to the team by the referee agent. The team agent may request player agents to perform searches of positions to certain depths.

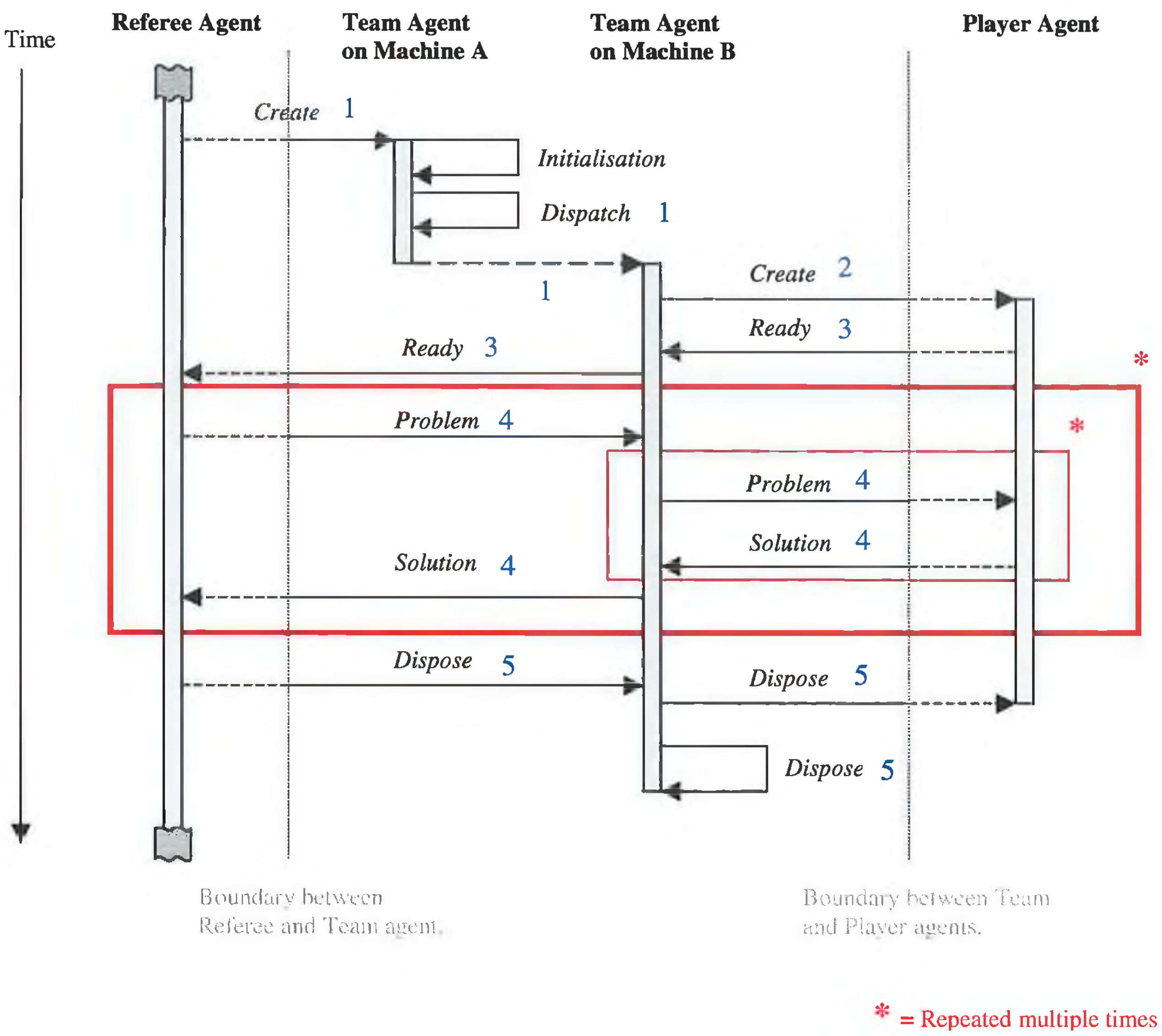



**Figure 5.4:** Time flow diagram of Team agent's interactions.

The following steps detail the actions of the team agent as indicated in Figure 5.4:

1) A referee agent creates the team agent. Following creation the first task of the team agent is to transport itself to the location specified by the referee agent.
2) When the team agent arrives at its new location, it then creates each of the player agents that the team was assigned. During creation the team agent

passes the player agents' personality information and the location that they should dispatch to.

3) The team agent receives ready messages from the player agents. When all these ready messages are received the team agent then sends a ready message to the referee agent.
4) When it is the team side's turn to move, it will receive board positions from the referee agent. For each of these board positions the team agent will request its player agent to perform searches to different depths. After all the searches are complete the team agent will decide on the team side's move and send it to the referee agent.
5) When the team agent receives the dispose message from the referee agent, the team agent first sends dispose messages to all its player agents and then disposes of itself.

*Player Agent*

The role of a player agent is to perform searches on behalf of its team agent. The team agent sends the player agent the position and depth to search. When the player agent is finished the search, it sends the PV and score of the search to its team agent.

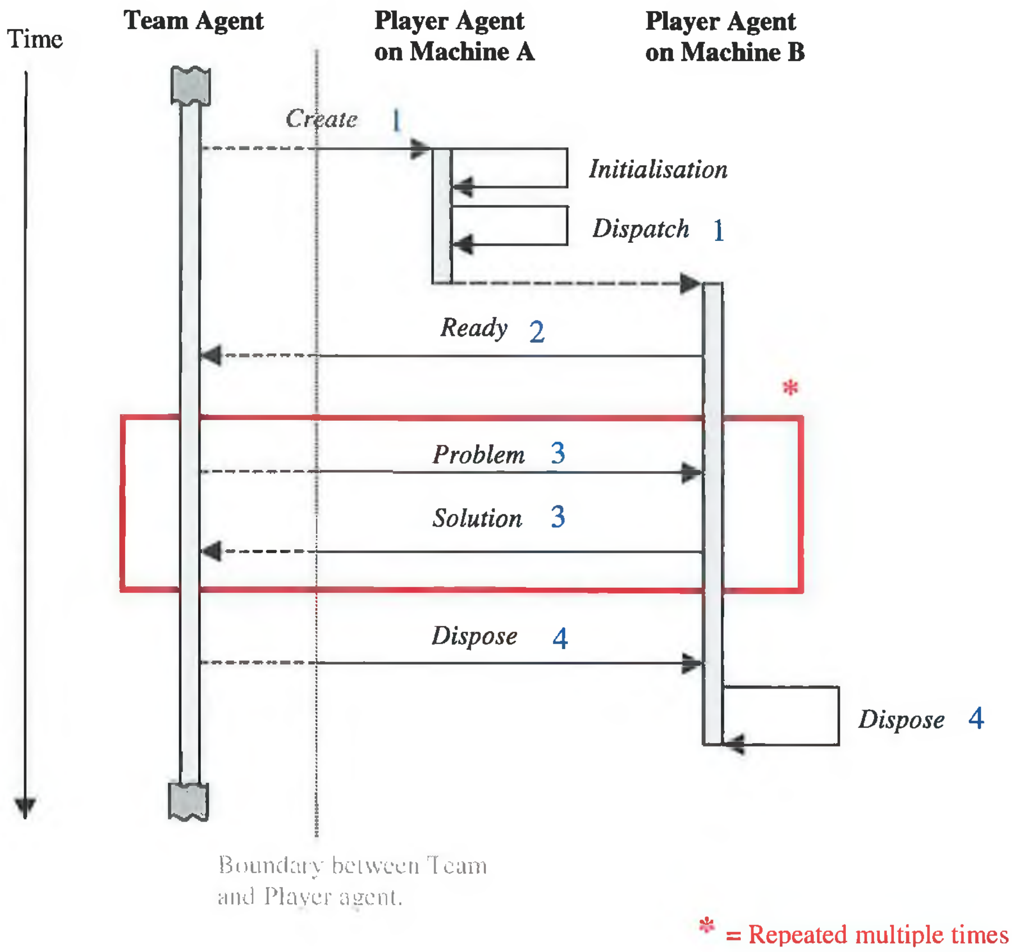



**Figure 5.5:** Time flow diagram of Player agent's interactions.

The following steps detail the actions in Figure 5.5:

1) A team agent creates the player agent. Among the information passed to the player agent is the location for it to reside on. The player agent dispatches itself to its specified location.
2) On arrival at its location the player agent sends a ready message to its team agent indicating its arrival.
3) The player agent receives position and depths to search. When finished a search it returns the PV and score of the search to its team agent.

4) When a dispose message is received from the team agent the player agent disposes of itself.

## 5.3 Chess Engine

Each of the player agents have their own instance of the chess engine. When a player invokes their instance of the chess engine it will have no effect on the other instances of the chess engine in the test system. All the instances of the chess engine use the same code base. The chess engine in the test system provides a means of performing the searches of the game tree. Given the initial position in the game tree to search from, the depth to search the position to and the side to move, the chess engine will search the tree and return the principle variation and its score. Each player agent has a personality associated with it using evaluation function parameters. When invoked, the chess engine extracts this information and uses it in its evaluation function. In this way the personality of a player agent is transferred to the chess engine.

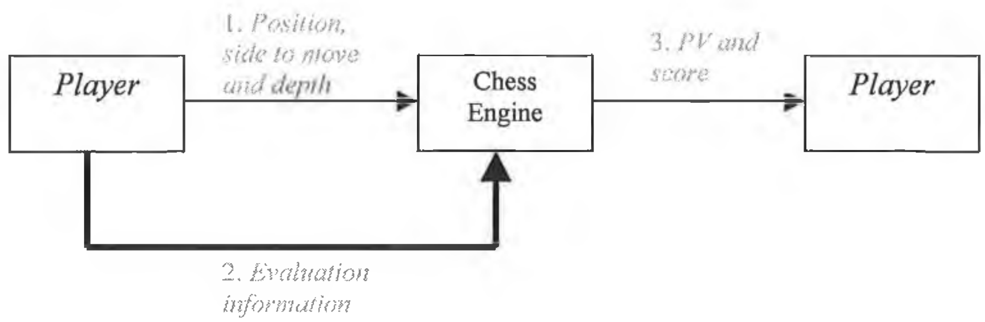



**Figure 5.6:** Series of steps that are performed by player agent and chess engine for search of position.

### 5.3.1 Search Algorithm

As discussed in Chapter 2, the search algorithm is the method used by a chess program to traverse a chess game tree. The search algorithm is responsible for expanding the chess game tree and collecting the information from the game tree terminals.

The search algorithm used in the test system's chess engine is the alpha-beta search algorithm. Quiescence search is implemented in the chess engine to allow terminal positions, which would normally be evaluated, to be expanded further to a certain depth. The search algorithm also implements simple move ordering by promoting capture and promotion moves to the top of a position's move list.

### 5.3.2 Evaluation Function

The purpose of the evaluation function is to estimate the value of a board for a given side. It performs this task by examining the board for positive and negative board features and calculates the score for the board from the payoffs associated with these features.

Crafty [Crafty URL] and TSCP [TSCP URL] are two computer chess programs whose source code is publicly available. The board features examined for by the test system have been assembled from a combination of the evaluation functions of Crafty 12.6 and TSCP 1.7. Not all features from these programs were implemented because of time constraints. The different features examined for by the evaluation function can be broken down into the following categories:

**Material Values**

*Material values are the payoffs associated with each of the pieces on the board.*

*Elements:*

- Pawn Value
- Rook Value
- Knight Value
- Bishop Value
- Queen Value

**Piece-Square Matrices**

*For each piece on the board there is also a payoff associated with their position on the board, which is found in the piece-square matrices. Payoffs can be either positive or negative depending on the piece's position on the board. For example, a knight piece may have a positive piece-square payoff if it is in the centre of the board where it would have more influence over the game. Alternatively, if the knight piece was at the edge of the board then its piece-square payoff may be negative as the piece would have little influence over the game and would be more easily trapped.*

*Elements:*

- Pawn Piece-Square Matrix
- Knight Piece-Square Matrix
- Bishop Piece-Square Matrix
- Rook Piece-Square Matrix
- Queen Piece-Square Matrix

**Pawn Features**

*Pawns features are penalties and bonuses associated with the pawn structure.*

*Elements:*

- Doubled Pawn Penalty
- Isolated Pawn Penalty
- Backward Pawn Penalty
- Passed Pawn Bonus
- Pawn Majority Queen Side

**Rook Features**

*The rook piece is especially influential on a game when it is on an open/semi-open file or on the $7^{th}$ row of the board.*

*Elements:*

- Rook Semi-Open File Bonus
- Rook Open File Bonus
- Rook On $7^{th}$ Row Bonus

**Knight Features**

*A knight outpost is a knight piece located in a centre square of the board and which is not attacked by any the opponent's pawn pieces. From a knight outpost the knight piece is extremely powerful in attack. The influence of a knight outpost is furthered if its square is supported by a pawn piece of its own.*

*Elements:*

- Knight Outpost Bonus

**King Safety**

*King safety refers to board features associated with king safety such as the position of pawns, threatening pieces and open files. The Protecting Bishop Bonus refers to the presence of a bishop piece that protects squares left after certain pawns have been advanced.*

*Elements:*

- Pawn Attack Penalty
- Rook's Pawn Position
- Knight's Pawn Position
- Bishop's Pawn Position
- Threatening Bishop Penalty
- Threat on g2 and Related Squares Penalty
- Protecting Bishop Bonus
- Open File Penalty

**Bishop Features**

*Mobility is essential to the bishop piece if it is to attack and defend. Each square that a bishop can move to is associated with a bishop mobility payoff. In special cases when the bishop is trapped, so the bishops mobility is zero, there is an extra penalty. If a side has both bishops on the board then it receives a bonus for this.*

*Elements:*

- Bishop Mobility
- Bishop Trapped Penalty
- Bishop Pair Bonus

**Piece-King Tropism**

*Tropism relates to the distance between a piece and the opponent's king. The closer the piece is to the king the higher the payoff for the feature.*

*Elements:*

- Knight-King Tropism
- Bishop-King Tropism
- Rook-King Tropism
- Queen-King Tropism

## 5.4 Critique of Test System

The design of the test system is based on the Aglets framework. The main purpose of using Aglets was to allow easy distribution of processing thus speeding up the testing process. Aglets are Java based software agents that can transport themselves and each other to different machine locations without interrupting their current execution state. In the design for the test system, team and player agents are created at one machine location before being sent to another machine location. At this second machine location the agent resides for the rest of its lifetime until it is destroyed. Thus the ability of aglets to transport themselves from one location to another is not really fundamental to the design of the test system. Other means of process distribution such as a simple client/server architecture would have served the purpose just as well. Such a client/server architecture would have greatly simplified the design and implementation of the test system.

The test system was not properly designed to handle the problem of repeated moves properly. In the game of chess, if the same move on the same board is repeated three times during a game then the game is declared a draw. The test system was designed to send new board positions from the referee to the team agents and from the team agents to the player agents. The system design does not take into account the previous moves and positions of the game. Without this information the team and player agents cannot tell if the move they choose will result in the game being declared drawn because of move repetition. The design of system should have included this aspect of the game of chess.

# Chapter 6

# Results and Analysis

In this chapter the details, methods and results of testing shall be explained. It will first begin by detailing the test data used and why. This will be followed by details of the various personalities used in testing and then by the measurements used to quantify the results. Finally in chronological order, details of tests and analysis of their results shall be given.

## 6.1 Construction of Test Set

Before testing can begin, test data must be first decided upon. In chess this translates into deciding upon different board positions that the system will play. Using the standard opening board position for tests is impractical, as the same game would just be repeated over and over. The use of depth control ensures this. To ensure the testing is of a practical nature, the board positions used should be found in master chess play. Uneven board positions will result in unfair results.

The beginning of chess games is dominated by chess opening sequences. These sequences have been found, over the centuries of human chess, to be "optimum play". The different sequences result in board positions that demand different styles of play. By choosing test data positions from different opening sequences one can obtain positions that are both diverse and occur in master chess play.

Opening sequences are broken down into different opening sequence types. The test data decided upon comprised of 24 opening sequences, consisting of 2 opening sequences from 12 different opening sequence types [Korn 1972]. The 12 different opening sequence types used were:

Tend to lead to open positions and tactical aggressive positions:

1) Scotch
2) Ruy Lopez
3) Centre

Semi-Open:

4) French Defence (Defensive for black)
5) Sicilian Defence (Semi-open for both white and black)

Semi-closed:

6) Queens Gambit Accepted
7) Queens Gambit Declined
8) Queens Pawn Games
9) Nimzo Indian Defence

Closed, tends to lead to closed positional positions:

10) Queens Indian Defence
11) Kings Indian Defence
12) Reti Opening

These different opening sequence types were chosen as they provided a broad range of positions that reflect different styles of play (the full details of the opening sequences are listed in Appendix C).

## 6.2 Details of Test Personalities

The objective of this research is to discover if multiple styles of play collaborating outperform that of a single style of play. If the research is to have any relevance the styles of play collaborating need to be different. Style of play in computer chess translates into different evaluation functions (see Section 4.3). Evaluation functions can be made different by changing the payoffs associated with the positive and negative board features that the evaluation function checks for.

Five different personalities were used for testing. These were:

1) Normal Personality
2) Aggressive Personality
3) Defensive Personality
4) Semi-Open Personality, and
5) Positional Personality

### 6.2.1 Normal Personality

The Normal personality can be viewed as the standard evaluation function used in single evaluation function programs, which includes most computer chess programs. The style of play of the personality is average; the personality tries to balance out all possible styles of play. It is neither too aggressive, defensive, semi-open nor positional but somewhere in between all of these.

The evaluation function of the test system was created from a combination of the evaluation functions of Crafty [Crafty URL] and TSCP [TSCP URL], as discussed in Section 5.3.2. To incorporate their average style of play into the test system's Normal personality, the weightings from their evaluation functions need to be transferred into the evaluation function of the Normal personality. The weightings used in the programs, however, are on a different scale. For example, the value for a queen piece in Crafty is 9000 compared to a value of 900 in TSCP. In order to transfer both sets of weightings into the Normal personality, the weightings of the evaluation functions need to be normalised on a common feature. This will insure that the weighting given to a feature maintains its intended influence on board position evaluation.

The weightings from the evaluation functions of Crafty and TSCP were normalised based on pawn value to create the weightings for the evaluation function of the Normal personality. The reason for this was that pawn value was present in both evaluation functions and because pawn value is commonly used in chess evaluation functions as the base value from which all other weightings are decided.

| | Crafty | TSCP | Test System |
|---|---|---|---|
| Pawn Value | **1000** | **100** | **1000** |
| Backwards Pawn Penalty | | **8** | **80** |
| Knight Outpost | **50** | | **50** |

**Table 6.1:** Conversion of elements from Crafty and TSCP into test system.

A list of board features and their payoffs in Crafty, TSCP and the test system are given in Table 6.1. The elements of Crafty and TSCP in the test system are normalised based on Crafty's pawn value. TSCP's pawn value is 10 times smaller than Crafty's. Therefore if an element from TSCP is to be included in the test system then 10 must multiply its payoff. The payoff of a feature in Crafty is directly transferable to the test system.

The evaluation function created by combining the evaluation functions of Crafty and TSCP is referred to in the test system as the Normal Personality and represents an average style of play. The other personalities in the test system will be from the perspective of this personality.

It is not possible to create the other personalities of the test system using the same composition mechanism used to create the Normal personality. If the style of play of both Crafty and TSCP were similar, which they should be, then it doesn't matter what element is used to normalise their weightings to create the Normal personality. The relative weightings of the evaluation function elements should be the same and so the Normal personality's style of play should be maintained. It is not the actual values in the evaluation function that are important for style of play but their relative values.

The Normal personality does not play the game of chess using multiple styles of play just because the weightings for its evaluation function come from different chess programs. The Normal personality has only one set of weightings in its evaluation function and so views the game tree from only one style of chess play.

A multiple player team may be able to play the game using multiple styles of play because as a team it would have multiple different evaluation functions and so would be viewing the chess game tree from multiple styles of play.

The features and payoffs of the Normal personality are given in Figure 6.2. The features are broken down into the categories described in Section 5.3.1.

*Pawn Features:*

DOUBLED_PAWN_PENALTY = 50
ISOLATED_PAWN_PENALTY = 100
BACKWARDS_PAWN_PENALTY = 80
PASSED_PAWN_BONUS = 40
PAWN_MAJORITY_QUEEN_SIDE = 100

*Rook Features:*

ROOK_SEMI_OPEN_FILE_BONUS = 25
ROOK_OPEN_FILE_BONUS = 100
ROOK_ON_SEVENTH_BONUS = 200

*Knight Features:*

KNIGHT_OUTPOST = 50

*Bishop Features:*

BISHOP_TRAPPED = 1500
BISHOP_MOBILITY = 15
BISHOP_PAIR = 100

*King Safety:*

KING_SAFETY_PAWN_ATTACK = 3
KING_SAFETY_RP_ADV1 = 1
KING_SAFETY_RP_ADV2 = 3
KING_SAFETY_RP_TOO_FAR = 4
KING_SAFETY_RP_MISSING = 5
KING_SAFETY_RP_FILE_OPEN = 5
KING_SAFETY_NP_ADV1 = 2
KING_SAFETY_NP_ADV2 = 4
KING_SAFETY_NP_TOO_FAR = 5
KING_SAFETY_NP_MISSING = 5
KING_SAFETY_NP_FILE_OPEN = 5
KING_SAFETY_BP_ADV1 = 1
KING_SAFETY_BP_ADV2 = 2
KING_SAFETY_BP_TOO_FAR = 3
KING_SAFETY_BP_MISSING = 3
KING_SAFETY_BP_FILE_OPEN = 2
KING_SAFETY_MATE_G2G7 = 10
KING_SAFETY_GOOD_BISHOP = 5
KING_SAFETY_OPEN_FILE = 5

*Piece-King Tropism:*

KNIGHT_KING_TROPISM = 12
BISHOP_KING_TROPISM = 8
ROOK_KING_TROPISM = 8
QUEEN_KING_TROPISM = 16

*Material Values:*

Pawn Value = 1000
Rook Value = 5000
Knight Value = 3300
Bishop Value = 3300
Queen Value = 9500

*Piece-Square Matrices:*

pawn_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -60 | -60 | 30 | 60 | 60 | -60 | -300 | -300 |
| -50 | -50 | 25 | 50 | 50 | -50 | -250 | -250 |
| -40 | -40 | 20 | 40 | 40 | -40 | -200 | -200 |
| -30 | -30 | 15 | 30 | 30 | -30 | -150 | -150 |
| -20 | -20 | 10 | 20 | 20 | -20 | -100 | -100 |
| -10 | -10 | 5 | 10 | 10 | -10 | -50 | -50 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

knight_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | 0 | 50 | 50 | 50 | 50 | 0 | -50 |
| -50 | 0 | 50 | 100 | 100 | 50 | 0 | -50 |
| -50 | 0 | 50 | 100 | 100 | 50 | 0 | -50 |
| -50 | 0 | 50 | 50 | 50 | 50 | 0 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |

bishop_pcsq

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | 0 | 50 | 50 | 50 | 50 | 0 | -50 |
| -50 | 0 | 50 | 100 | 100 | 50 | 0 | -50 |
| -50 | 0 | 50 | 100 | 100 | 50 | 0 | -50 |
| -50 | 0 | 50 | 50 | 50 | 50 | 0 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |

rook_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |

queen_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 40 | 40 | 40 | 40 | 40 | 40 | 0 |
| 0 | 40 | 80 | 80 | 80 | 80 | 40 | 0 |
| 0 | 40 | 80 | 120 | 120 | 80 | 40 | 0 |
| 0 | 40 | 80 | 120 | 120 | 80 | 40 | 0 |
| 0 | 40 | 80 | 80 | 80 | 80 | 40 | 0 |
| 0 | 40 | 40 | 40 | 40 | 40 | 40 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6.2:** Features and payoffs for Normal personality.

### 6.2.2 Aggressive Personality

The aggressive personality is more interested in gaining attacking advantages than the normal personality. Movement of pieces is crucial when attacking and this is reflected in the increased importance of the mobility of the bishop pieces, knight outposts and bishop, knight and queen pieces securing squares in the centre of the board. When attacking, the aggressive personality wants to get its pieces as close to the enemy king as possible, which is reflected in the increase in the piece-king tropism values. The defensive considerations of pawns are of less importance to the aggressive personality as demonstrated by the decrease in pawn penalty values. Rooks are crucial in attacks and their importance is reflected in the increased bonus for rooks on open/semi-open files, and along the kingside files. The bonus for a rook on the $7^{th}$ rank is also significantly increased. Pawns are of less importance to an aggressive personality so their material value decreases while bishop pieces provide many attacking possibilities reflected in their increased material value.

The payoffs associated with the Aggressive personality are given in Figure 6.3. The differences between the Aggressive personality and the Normal personality are highlighted in bold.

*Pawn Features:*

**DOUBLED_PAWN_PENALTY = 25**
**ISOLATED_PAWN_PENALTY = 50**
**BACKWARDS_PAWN_PENALTY = 40**
PASSED_PAWN_BONUS = 40
PAWN_MAJORITY_QUEEN_SIDE = 100

*Rook Features:*

**ROOK_SEMI_OPEN_FILE_BONUS = 125**
**ROOK_OPEN_FILE_BONUS = 500**
**ROOK_ON_SEVENTH_BONUS = 1000**

*Knight Features:*

**KNIGHT_OUTPOST = 100**

*Bishop Features:*

BISHOP_TRAPPED = 1500
**BISHOP_MOBILITY = 30**
BISHOP_PAIR = 100

*King Safety:*

KING_SAFETY_PAWN_ATTACK = 3
KING_SAFETY_RP_ADV1 = 1
KING_SAFETY_RP_ADV2 = 3
KING_SAFETY_RP_TOO_FAR = 4
KING_SAFETY_RP_MISSING = 5
KING_SAFETY_RP_FILE_OPEN = 5
KING_SAFETY_NP_ADV1 = 2
KING_SAFETY_NP_ADV2 = 4
KING_SAFETY_NP_TOO_FAR = 5
KING_SAFETY_NP_MISSING = 5
KING_SAFETY_NP_FILE_OPEN = 5
KING_SAFETY_BP_ADV1 = 1
KING_SAFETY_BP_ADV2 = 2
KING_SAFETY_BP_TOO_FAR = 3
KING_SAFETY_BP_MISSING = 3
KING_SAFETY_BP_FILE_OPEN = 2
KING_SAFETY_MATE_G2G7 = 10
KING_SAFETY_GOOD_BISHOP = 5
KING_SAFETY_OPEN_FILE = 5

*Piece-King Tropism:*

**KNIGHT_KING_TROPISM = 48**
**BISHOP_KING_TROPISM = 32**
**ROOK_KING_TROPISM = 32**
**QUEEN_KING_TROPISM = 64**

*Material Values:*

**Pawn Value = 850**
Rook Value = 5000
Knight Value = 3300
**Bishop Value = 4000**
Queen Value = 9500

*Piece-Square Matrices:*

pawn_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -60 | -60 | 30 | 60 | 60 | -60 | -300 | -300 |
| -50 | -50 | 25 | 50 | 50 | -50 | -250 | -250 |
| -40 | -40 | 20 | 40 | 40 | -40 | -200 | -200 |
| -30 | -30 | 15 | 30 | 30 | -30 | -150 | -150 |
| -20 | -20 | 10 | 20 | 20 | -20 | -100 | -100 |
| -10 | -10 | 5 | 10 | 10 | -10 | -50 | -50 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

knight_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | 0 | 75 | 75 | 75 | 75 | 50 | -50 |
| -50 | 0 | 75 | 200 | 200 | 150 | 100 | -50 |
| -50 | 0 | 75 | 200 | 200 | 100 | 50 | -50 |
| -50 | 0 | 75 | 75 | 75 | 75 | 50 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |

bishop_pcsq

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | 0 | 75 | 75 | 75 | 75 | 0 | -50 |
| -50 | 0 | 75 | 150 | 150 | 75 | 0 | -50 |
| -50 | 0 | 75 | 150 | 150 | 75 | 0 | -50 |
| -50 | 0 | 75 | 75 | 75 | 75 | 0 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |

rook_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 40 | 80 | 100 | 100 | 100 | 75 | 40 |
| 0 | 40 | 80 | 100 | 100 | 100 | 75 | 40 |
| 0 | 40 | 80 | 100 | 100 | 100 | 75 | 40 |
| 0 | 40 | 80 | 100 | 100 | 100 | 75 | 40 |
| 0 | 40 | 80 | 100 | 100 | 100 | 75 | 40 |
| 0 | 40 | 80 | 100 | 100 | 100 | 75 | 40 |
| 0 | 40 | 80 | 100 | 100 | 100 | 75 | 40 |
| 0 | 40 | 80 | 100 | 100 | 100 | 75 | 40 |

queen_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 60 | 60 | 60 | 60 | 60 | 60 | 0 |
| 0 | 60 | 120 | 120 | 120 | 120 | 60 | 0 |
| 0 | 60 | 120 | 180 | 180 | 120 | 60 | 0 |
| 0 | 60 | 120 | 180 | 180 | 120 | 60 | 0 |
| 0 | 60 | 120 | 120 | 120 | 120 | 60 | 0 |
| 0 | 60 | 60 | 60 | 60 | 60 | 60 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6.3:** Features and payoffs for Aggressive personality.

### 6.2.3 Defensive Personality

The Defensive personality's priority is to defend the king and maintain pawn formation. The importance of king defence is increased by the increased penalties related to king safety. The increase of importance of defensive pawn features is demonstrated by the larger pawn penalty values. Attacking the opponent's king is of less importance to the Defensive personality than it is to the Normal personality; therefore the values associated with piece-king tropism have been decreased.

The payoffs associated with the Defensive personality are given in Figure 6.4. The differences between the Defensive personality and the Normal personality are highlighted in bold.

*Pawn Features:*

**DOUBLED_PAWN_PENALTY = 100**
**ISOLATED_PAWN_PENALTY = 200**
**BACKWARDS_PAWN_PENALTY = 160**
PASSED_PAWN_BONUS = 40
PAWN_MAJORITY_QUEEN_SIDE = 100

*Rook Features:*

ROOK_SEMI_OPEN_FILE_BONUS = 25
ROOK_OPEN_FILE_BONUS = 100
ROOK_ON_SEVENTH_BONUS = 200

*Knight Features:*

KNIGHT_OUTPOST = 50

*Bishop Features:*

BISHOP_TRAPPED = 1500
BISHOP_MOBILITY = 15
BISHOP_PAIR = 100

*King Safety:*

**KING_SAFETY_PAWN_ATTACK = 15**
**KING_SAFETY_RP_ADV1 = 5**
**KING_SAFETY_RP_ADV2 = 15**
**KING_SAFETY_RP_TOO_FAR = 20**
**KING_SAFETY_RP_MISSING = 25**
**KING_SAFETY_RP_FILE_OPEN = 25**
**KING_SAFETY_NP_ADV1 = 10**
**KING_SAFETY_NP_ADV2 = 20**
**KING_SAFETY_NP_TOO_FAR = 25**
**KING_SAFETY_NP_MISSING = 25**
**KING_SAFETY_NP_FILE_OPEN = 25**
**KING_SAFETY_BP_ADV1 = 5**
**KING_SAFETY_BP_ADV2 = 10**
**KING_SAFETY_BP_TOO_FAR = 15**
**KING_SAFETY_BP_MISSING = 15**
**KING_SAFETY_BP_FILE_OPEN = 10**
**KING_SAFETY_MATE_G2G7 = 50**
**KING_SAFETY_GOOD_BISHOP = 25**
**KING_SAFETY_OPEN_FILE = 25**

*Piece-King Tropism:*

**KNIGHT_KING_TROPISM = 6**
**BISHOP_KING_TROPISM = 4**
**ROOK_KING_TROPISM = 4**
**QUEEN_KING_TROPISM = 8**

*Material Values:*

Pawn Value = 1000
Rook Value = 5000
Knight Value = 3300
Bishop Value = 3300
Queen Value = 9500

*Piece-Square Matrices:*

pawn_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -60 | -60 | 30 | 60 | 60 | -60 | -300 | -300 |
| -50 | -50 | 25 | 50 | 50 | -50 | -250 | -250 |
| -40 | -40 | 20 | 40 | 40 | -40 | -200 | -200 |
| -30 | -30 | 15 | 30 | 30 | -30 | -150 | -150 |
| -20 | -20 | 10 | 20 | 20 | -20 | -100 | -100 |
| -10 | -10 | 5 | 10 | 10 | -10 | -50 | -50 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

knight_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | 0 | 50 | 50 | 50 | 50 | 0 | -50 |
| -50 | 0 | 50 | 100 | 100 | 50 | 0 | -50 |
| -50 | 0 | 50 | 100 | 100 | 50 | 0 | -50 |
| -50 | 0 | 50 | 50 | 50 | 50 | 0 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |

bishop_pcsq

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | 0 | 50 | 50 | 50 | 50 | 0 | -50 |
| -50 | 0 | 50 | 100 | 100 | 50 | 0 | -50 |
| -50 | 0 | 50 | 100 | 100 | 50 | 0 | -50 |
| -50 | 0 | 50 | 50 | 50 | 50 | 0 | -50 |
| -50 | 0 | 0 | 0 | 0 | 0 | 0 | -50 |
| -50 | -50 | -50 | -50 | -50 | -50 | -50 | -50 |

rook_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |
| 0 | 40 | 80 | 100 | 100 | 80 | 40 | 0 |

queen_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 40 | 40 | 40 | 40 | 40 | 40 | 0 |
| 0 | 40 | 80 | 80 | 80 | 80 | 40 | 0 |
| 0 | 40 | 80 | 120 | 120 | 80 | 40 | 0 |
| 0 | 40 | 80 | 120 | 120 | 80 | 40 | 0 |
| 0 | 40 | 80 | 80 | 80 | 80 | 40 | 0 |
| 0 | 40 | 40 | 40 | 40 | 40 | 40 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6.4:** Features and payoffs for Defensive Personality.

### 6.2.4 Semi-Open Personality

A semi-open personality is aggressive in nature and likes to control the game board via piece mobility and positional advantages. The ability to move the focus of attack from one part of the board to another, quickly and easily, is crucial to the personality. This ability is achieved by the increased importance of pieces being in the centre exerting their influence over both sides of the board. The importance of piece mobility is reflected in the increased importance of bishop mobility and knight outposts. The semi-open personality likes rooks on open or semi-open files where attacks can come quickly and decisively as reflected in their increased payoffs. Rooks on the $7^{th}$ rank are attacking in nature and have increased importance to the semi-open personality.

The payoffs associated with the semi-open personality are given in Figure 6.5. The difference between the semi-open personality and the Normal personality are highlighted in bold.

*Pawn Features:*

DOUBLED_PAWN_PENALTY = 50
ISOLATED_PAWN_PENALTY = 100
BACKWARDS_PAWN_PENALTY = 80
PASSED_PAWN_BONUS = 40
PAWN_MAJORITY_QUEEN_SIDE = 100

*Rook Features:*

**ROOK_SEMI_OPEN_FILE_BONUS = 75**
**ROOK_OPEN_FILE_BONUS = 300**
**ROOK_ON_SEVENTH_BONUS = 600**

*Knight Features:*

**KNIGHT_OUTPOST = 100**

*Bishop Features:*

BISHOP_TRAPPED = 1500
**BISHOP_MOBILITY = 60**
BISHOP_PAIR = 100

*King Safety:*

KING_SAFETY_PAWN_ATTACK = 3
KING_SAFETY_RP_ADV1 = 1
KING_SAFETY_RP_ADV2 = 3
KING_SAFETY_RP_TOO_FAR = 4
KING_SAFETY_RP_MISSING = 5
KING_SAFETY_RP_FILE_OPEN = 5
KING_SAFETY_NP_ADV1 = 2
KING_SAFETY_NP_ADV2 = 4
KING_SAFETY_NP_TOO_FAR = 5
KING_SAFETY_NP_MISSING = 5
KING_SAFETY_NP_FILE_OPEN = 5
KING_SAFETY_BP_ADV1 = 1
KING_SAFETY_BP_ADV2 = 2
KING_SAFETY_BP_TOO_FAR = 3
KING_SAFETY_BP_MISSING = 3
KING_SAFETY_BP_FILE_OPEN = 2
KING_SAFETY_MATE_G2G7 = 10
KING_SAFETY_GOOD_BISHOP = 5
KING_SAFETY_OPEN_FILE = 5

*Piece-King Tropism:*

KNIGHT_KING_TROPISM = 12
BISHOP_KING_TROPISM = 8
ROOK_KING_TROPISM = 8
QUEEN_KING_TROPISM = 16

*Material Values:*

Pawn Value = 1000
Rook Value = 5000
Knight Value = 3300
Bishop Value = 3300
Queen Value = 9500

*Piece-Square Matrices:*

pawn_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -60 | -60 | 30 | 60 | 60 | -60 | -300 | -300 |
| -50 | -50 | 25 | 50 | 50 | -50 | -250 | -250 |
| -40 | -40 | 20 | 40 | 40 | -40 | -200 | -200 |
| -30 | -30 | 15 | 30 | 30 | -30 | -150 | -150 |
| -20 | -20 | 10 | 20 | 20 | -20 | -100 | -100 |
| -10 | -10 | 5 | 10 | 10 | -10 | -50 | -50 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

knight_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -100 | -100 | -100 | -100 | -100 | -100 | -100 | -100 |
| -100 | 0 | 0 | 0 | 0 | 0 | 0 | -100 |
| -100 | 0 | 100 | 100 | 100 | 100 | 0 | -100 |
| -100 | 0 | 100 | 200 | 200 | 100 | 0 | -100 |
| -100 | 0 | 100 | 200 | 200 | 100 | 0 | -100 |
| -100 | 0 | 100 | 100 | 100 | 100 | 0 | -100 |
| -100 | 0 | 0 | 0 | 0 | 0 | 0 | -100 |
| -100 | -100 | -100 | -100 | -100 | -100 | -100 | -100 |

bishop_pcsq

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -150 | -150 | -150 | -150 | -150 | -150 | -150 | -150 |
| -150 | 0 | 0 | 0 | 0 | 0 | 0 | -150 |
| -150 | 0 | 150 | 150 | 150 | 150 | 0 | -150 |
| -150 | 0 | 150 | 300 | 300 | 150 | 0 | -150 |
| -150 | 0 | 150 | 300 | 300 | 150 | 0 | -150 |
| -150 | 0 | 150 | 150 | 150 | 150 | 0 | -150 |
| -150 | 0 | 0 | 0 | 0 | 0 | 0 | -150 |
| -150 | -150 | -150 | -150 | -150 | -150 | -150 | -150 |

rook_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 120 | 240 | 300 | 300 | 240 | 120 | 0 |
| 0 | 120 | 240 | 300 | 300 | 240 | 120 | 0 |
| 0 | 120 | 240 | 300 | 300 | 240 | 120 | 0 |
| 0 | 120 | 240 | 300 | 300 | 240 | 120 | 0 |
| 0 | 120 | 240 | 300 | 300 | 240 | 120 | 0 |
| 0 | 120 | 240 | 300 | 300 | 240 | 120 | 0 |
| 0 | 120 | 240 | 300 | 300 | 240 | 120 | 0 |
| 0 | 120 | 240 | 300 | 300 | 240 | 120 | 0 |

queen_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 120 | 120 | 120 | 120 | 120 | 120 | 0 |
| 0 | 120 | 240 | 240 | 240 | 240 | 120 | 0 |
| 0 | 120 | 240 | 360 | 360 | 240 | 120 | 0 |
| 0 | 120 | 240 | 360 | 360 | 240 | 120 | 0 |
| 0 | 120 | 240 | 240 | 240 | 240 | 120 | 0 |
| 0 | 120 | 120 | 120 | 120 | 120 | 120 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6.5:** Features and payoffs for Semi-Open Personality.

### 6.2.5 Positional Personality

A positional personality concentrates on the accumulation of small positional advantages, over a long period, to gain control of more and more of the game board until the advantages overwhelm the opponent. The positional personality tries to accumulate these positional advantages without introducing any perceived long-term weaknesses in their position. Pawn structure is very important to a positional personality and this is reflected by the increased payoffs to the penalties and bonuses associated with pawn features. Positional advantages are gained by getting pieces into better squares of the board and hence controlling more important areas of the board. This importance of piece position is reflected in the increased magnitude of payoffs for the pieces-square matrices. Knight outposts are a reflection of a positional advantage and this is reflected in the payoffs. The positional advantage of a bishop is dependent on its mobility, which is reflected in the increased importance of bishop mobility to the personality.

The payoffs associated with the positional personality are given in Figure 6.6. The difference between the positional personality and the Normal personality are highlighted in bold.

*Pawn Features:*

**DOUBLED_PAWN_PENALTY = 200**
**ISOLATED_PAWN_PENALTY = 200**
**BACKWARDS_PAWN_PENALTY = 200**
**PASSED_PAWN_BONUS = 500**
**PAWN_MAJORITY_QUEEN_SIDE = 200**

*Rook Features:*

**ROOK_SEMI_OPEN_FILE_BONUS = 50**
**ROOK_OPEN_FILE_BONUS = 200**
**ROOK_ON_SEVENTH_BONUS = 400**

*Knight Features:*

**KNIGHT_OUTPOST = 200**

*Bishop Features:*

BISHOP_TRAPPED = 1500
**BISHOP_MOBILITY = 30**
BISHOP_PAIR = 100

*King Safety:*

KING_SAFETY_PAWN_ATTACK = 3
KING_SAFETY_RP_ADV1 = 1
KING_SAFETY_RP_ADV2 = 3
KING_SAFETY_RP_TOO_FAR = 4
KING_SAFETY_RP_MISSING = 5
KING_SAFETY_RP_FILE_OPEN = 5
KING_SAFETY_NP_ADV1 = 2
KING_SAFETY_NP_ADV2 = 4
KING_SAFETY_NP_TOO_FAR = 5
KING_SAFETY_NP_MISSING = 5
KING_SAFETY_NP_FILE_OPEN = 5
KING_SAFETY_BP_ADV1 = 1
KING_SAFETY_BP_ADV2 = 2
KING_SAFETY_BP_TOO_FAR = 3
KING_SAFETY_BP_MISSING = 3
KING_SAFETY_BP_FILE_OPEN = 2
KING_SAFETY_MATE_G2G7 = 10
KING_SAFETY_GOOD_BISHOP = 5
KING_SAFETY_OPEN_FILE = 5

*Piece-King Tropism:*

KNIGHT_KING_TROPISM = 12
BISHOP_KING_TROPISM = 8
ROOK_KING_TROPISM = 8
QUEEN_KING_TROPISM = 16

*Material Values:*

Pawn Value = 1000
Rook Value = 5000
Knight Value = 3300
Bishop Value = 3300
Queen Value = 9500

*Piece-Square Matrices:*

pawn_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **30** | **30** | 30 | 60 | 60 | -60 | **-200** | **-200** |
| **25** | **25** | 25 | 50 | 50 | -50 | **-150** | **-150** |
| **20** | **20** | 20 | 40 | 40 | -40 | **-120** | **-120** |
| **15** | **15** | 15 | 30 | 30 | -30 | **-100** | **-100** |
| **10** | **10** | 10 | 20 | 20 | -20 | **-50** | **-50** |
| **5** | **5** | 5 | 10 | 10 | -10 | **-25** | **-25** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

knight_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -100 | -100 | -100 | -100 | -100 | -100 | -100 | -100 |
| -100 | 0 | 0 | 0 | 0 | 0 | 0 | -100 |
| -100 | 0 | 100 | 100 | 100 | 100 | 0 | -100 |
| -100 | 0 | 100 | 200 | 200 | 100 | 0 | -100 |
| -100 | 0 | 100 | 200 | 200 | 100 | 0 | -100 |
| -100 | 0 | 100 | 100 | 100 | 100 | 0 | -100 |
| -100 | 0 | 0 | 0 | 0 | 0 | 0 | -100 |
| -100 | -100 | -100 | -100 | -100 | -100 | -100 | -100 |

bishop_pcsq

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -100 | -100 | -100 | -100 | -100 | -100 | -100 | -100 |
| -100 | 0 | 0 | 0 | 0 | 0 | 0 | -100 |
| -100 | 0 | 100 | 100 | 100 | 100 | 0 | -100 |
| -100 | 0 | 100 | 200 | 200 | 100 | 0 | -100 |
| -100 | 0 | 100 | 200 | 200 | 100 | 0 | -100 |
| -100 | 0 | 100 | 100 | 100 | 100 | 0 | -100 |
| -100 | 0 | 0 | 0 | 0 | 0 | 0 | -100 |
| -100 | -100 | -100 | -100 | -100 | -100 | -100 | -100 |

rook_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 80 | 160 | 200 | 200 | 160 | 80 | 0 |
| 0 | 80 | 160 | 200 | 200 | 160 | 80 | 0 |
| 0 | 80 | 160 | 200 | 200 | 160 | 80 | 0 |
| 0 | 80 | 160 | 200 | 200 | 160 | 80 | 0 |
| 0 | 80 | 160 | 200 | 200 | 160 | 80 | 0 |
| 0 | 80 | 160 | 200 | 200 | 160 | 80 | 0 |
| 0 | 80 | 160 | 200 | 200 | 160 | 80 | 0 |
| 0 | 80 | 160 | 200 | 200 | 160 | 80 | 0 |

queen_pcsq:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 80 | 80 | 80 | 80 | 80 | 80 | 0 |
| 0 | 80 | 160 | 160 | 160 | 160 | 80 | 0 |
| 0 | 80 | 160 | 240 | 240 | 160 | 80 | 0 |
| 0 | 80 | 160 | 240 | 240 | 160 | 80 | 0 |
| 0 | 80 | 160 | 160 | 160 | 160 | 80 | 0 |
| 0 | 80 | 80 | 80 | 80 | 80 | 80 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6.6:** Features and payoffs for Positional Personality.

## 6.3 Arrangement of Teams and Personalities

For testing, a standard multiple player team is required so that progress may be measured. The multiple player team used is a team made up of each of the various personalities. That is the multiple player team consists of a Normal personality, an Aggressive personality, a Defensive personality, a Semi-Open personality and a Positional personality. This multiple player team will be referred to as NADSP, each letter representing each personality.

The multiple player team requires standard opposition too. The opposition used is a team composed of just the Normal personality. The Normal personality is used as it is the personality based on the evaluation functions of Crafty and TSCP and so is likely to play sensible chess. The single player team will be referred to as N, the letter representing the personality in the team.

When a team receives a board position it will request each of its personalities to search the position. Since the multiple player team has more personalities than the single player team, the multiple player team has the advantage of performing more searches of the game tree. To compensate for this advantage the single player team is given the ability to search deeper than the multiple player team. The single player team can search to a maximum depth of 7-ply. The multiple player team, on the other hand, can only perform searches to depth 6. The number of 6-ply searches it can perform will also be controlled.

The chess engine of the test system uses an alpha-beta algorithm to search the game tree. An additional ply of search using the alpha-beta algorithm takes approximately 6.16 times the computational power of the previous search depth [Levy and Newborn 1991]. Using this measurement the relative computational power given to the teams can be approximated. Initially the single player team has an advantage, as the computational power to perform a 7-ply search is larger than that required for five 6-ply searches. The number and depth of searches will change with the MPSMs.

For tests, the multiple player team (NADSP) will be the black side and the single player team (N) will be white. The single player team, playing as white, will have a small advantage by moving first in the game of chess. In professional chess games since 1997 white has won 38% of the games, black 29% and 31% of the games have been drawn [ChessLab URL]. The games between the two sides will be refereed to as NvNADSP (single player team versus multiple player team).

### 6.4 Initial Method of Point Assignment

A means of quantifying the results from the chess games is required. A game of chess has three possible outcomes, which are a win for white, a win for black or a draw. Generally these results are quantified by giving a side one point for a win, zero points for a loss and a half point for a draw. This scoring measure will be used for quantifying the results from the test system.

The purpose of this work is to investigate the possible use of collaboration in computer chess. Therefore in the NvNADSP games, it is the results of NADSP that will interest us most. For this work the number of points black (NADSP) achieves will measure progress.

| Game Outcome | Points |
|---|---|
| White win | 0 |
| Black win | 1 |
| Draw | ½ |

**Table 6.7:** Point Assignment for Testing.

### 6.5 Description of MPSM 1

The purpose of the MPSMs is to combine and analysis the searches by the various players of the multiple player team and to resolve which of the players' recommended moves would be used as the team's next move. MPSM 1 is the initial MPSM created and uses a simple method to resolve this task. The steps for MPSM 1 are as follows:

1) The team agent initially sends out the new board position to the players in the team. Each of the players performs a 6-ply search of the position and returns its solution, consisting of the principle variation and score, to the team agent.

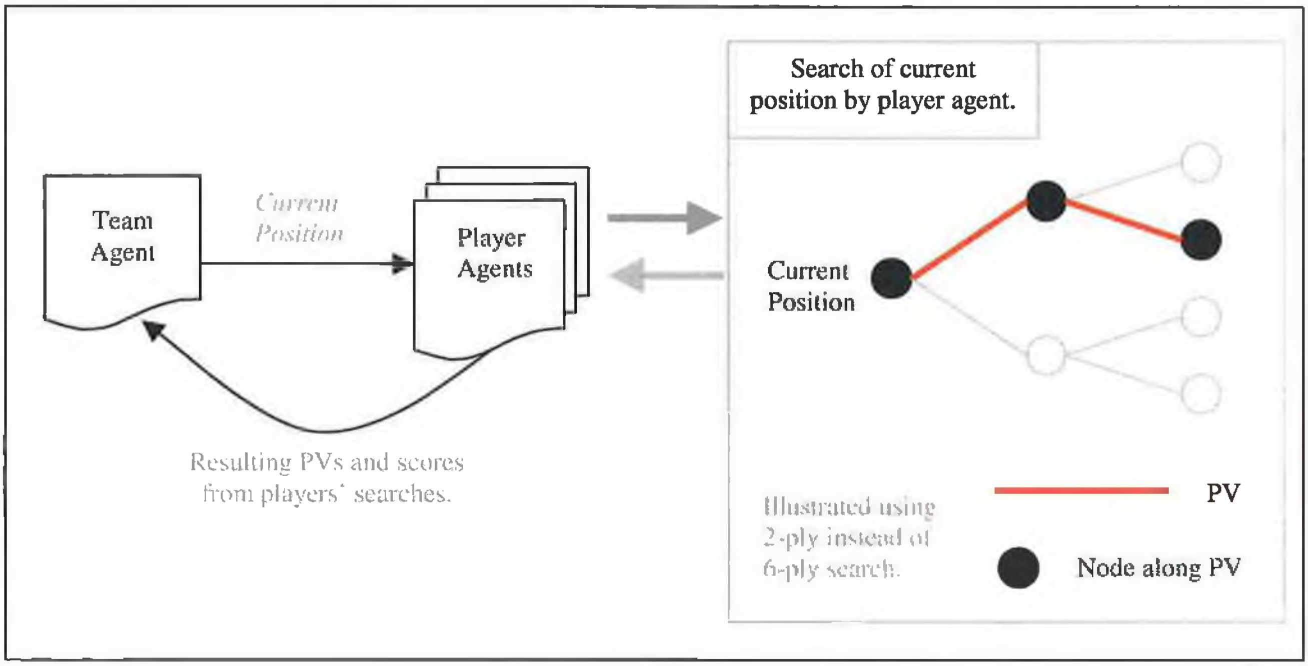
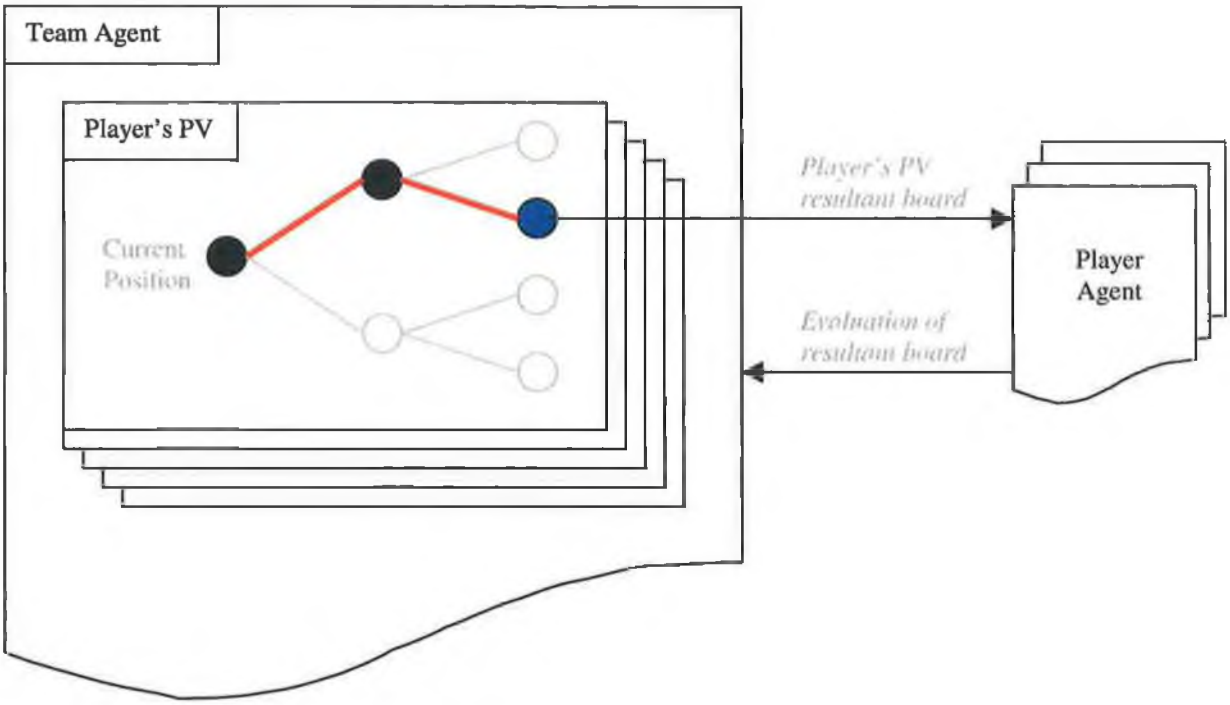



**Figure 6.8:** Step 1 of MPSM 1.

2) The resultant board positions from each of the players' initial search are sent to each of the player agents. The player agents evaluate the positions and return their evaluations back to the team agent. The team agent stores the scores for the resultant boards. To illustrate the remaining steps in MPSM 1, a sample table of possible resultant board player evaluations are given below.




| | Team Players' PVs | | | | |
|---|---|---|---|---|---|
| **Evaluations** | 1st Player's | 2nd Player's | 3rd Player's | 4th Player's | 5th Player's |
| 1st Player | +43 | +23 | -129 | +6 | -78 |
| 2nd Player | +3 | +53 | -15 | +30 | -44 |
| 3rd Player | -98 | +44 | +66 | -3 | +2 |
| 4th Player | -23 | -2 | -12 | -1 | -34 |
| 5th Player | +7 | -18 | +9 | -32 | +22 |

**Figure 6.9:** Step 2 of MPSM 1.

3) The players' PV resultant board evaluations are then ordered from best to worst.

| | Team Players' PVs | | | | |
|---|---|---|---|---|---|
| Evaluations | 1st Player's | 2nd Player's | 3rd Player's | 4th Player's | 5th Player's |
| 1st Player | 1st | 2nd | 5th | 3rd | 4th |
| 2nd Player | 3rd | 1st | 4th | 2nd | 5th |
| 3rd Player | 5th | 2nd | 1st | 4th | 3rd |
| 4th Player | 4th | 2nd | 3rd | 1st | 5th |
| 5th Player | 3rd | 4th | 2nd | 5th | 1st |

**Figure 6.10:** Step 3 of MPSM 1.

4) From the best to worst, the resultant boards are assigned a value from 0 to 4 respectively.

| | Team Players' PVs | | | | |
|---|---|---|---|---|---|
| Evaluations | 1st Player's | 2nd Player's | 3rd Player's | 4th Player's | 5th Player's |
| 1st Player | 0 | 1 | 4 | 2 | 3 |
| 2nd Player | 2 | 0 | 3 | 1 | 4 |
| 3rd Player | 4 | 1 | 0 | 3 | 2 |
| 4th Player | 3 | 1 | 2 | 0 | 4 |
| 5th Player | 2 | 3 | 1 | 4 | 0 |

**Figure 6.11:** Step 4 of MPSM 1.

5) The values of all the players' PVs are then summed. The players' PV that has the lowest score is selected as the team's move. In the example given, the 2nd player's PV has the lowest score with 6 points and so the first move in this PV will be used as the team's next move.

|  | Team Players' PVs |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Evaluations** | 1st Player's | 2nd Player's | 3rd Player's | 4th Player's | 5th Player's |
| 1st Player | 0 | 1 | 4 | 2 | 3 |
| 2nd Player | 2 | 0 | 3 | 1 | 4 |
| 3rd Player | 4 | 1 | 0 | 3 | 2 |
| 4th Player | 3 | 1 | 2 | 0 | 4 |
| 5th Player | 2 | 3 | 1 | 4 | 0 |
| Totals | 11 | 6 | 10 | 10 | 13 |

**Figure 6.12:** Step 5 of MPSM 1.

Performing tests using the NvNADSP teams set-up using MPSM 1 creates the following tables and results:

| No. | Opening Sequence | Result |
| --- | --- | --- |
| 1 | Centre - 1 | Draw |
| 2 | Centre - 2 | White |
| 3 | French Defence - 1 | White |
| 4 | French Defence - 2 | White |
| 5 | Kings Indian Defence - 1 | White |
| 6 | Kings Indian Defence - 2 | Draw |
| 7 | Nimzo Indian Defence - 1 | White |
| 8 | Nimzo Indian Defence - 2 | White |
| 9 | Queens Gambit Accepted - 1 | Draw |
| 10 | Queens Gambit Accepted - 2 | White |
| 11 | Queens Gambit Declined - 1 | Draw |
| 12 | Queens Gambit Declined - 2 | White |
| 13 | Queens Indian Defence - 1 | White |
| 14 | Queens Indian Defence - 2 | White |
| 15 | Queens Pawn Games - 1 | Draw |
| 16 | Queens Pawn Games - 2 | Draw |
| 17 | Reti Opening - 1 | Black |
| 18 | Reti Opening - 2 | White |
| 19 | Ruy Lopez - 1 | Draw |
| 20 | Ruy Lopez - 2 | White |
| 21 | Scotch - 1 | Black |
| 22 | Scotch - 2 | Draw |
| 23 | Sicilian Defence - 1 | Draw |
| 24 | Sicilian Defence - 2 | Draw |

| Result | Result Occurrence | Points per Result | Result Points |
|---|---|---|---|
| White Win | 12 | 0 | 0 |
| Black Win | 2 | 1 | 2 |
| Draw | 10 | 0.5 | 5 |
| | | **Total** | **7** |

**Test Setup 1:** Game results and scoring for MPSM 1.

### 6.6 Description of MPSM 2

MPSM 1 suffers from the setback of not taking into account the intermediate board positions between the current board and the PV resultant boards. MPSM 2 attempts to resolve this by updating the board position by just the first move in the PV and not the entire PV. These board positions are then searched to a depth of 5-ply. The result should be a more accurate representation of a player's evaluation of using a PV for the next move.

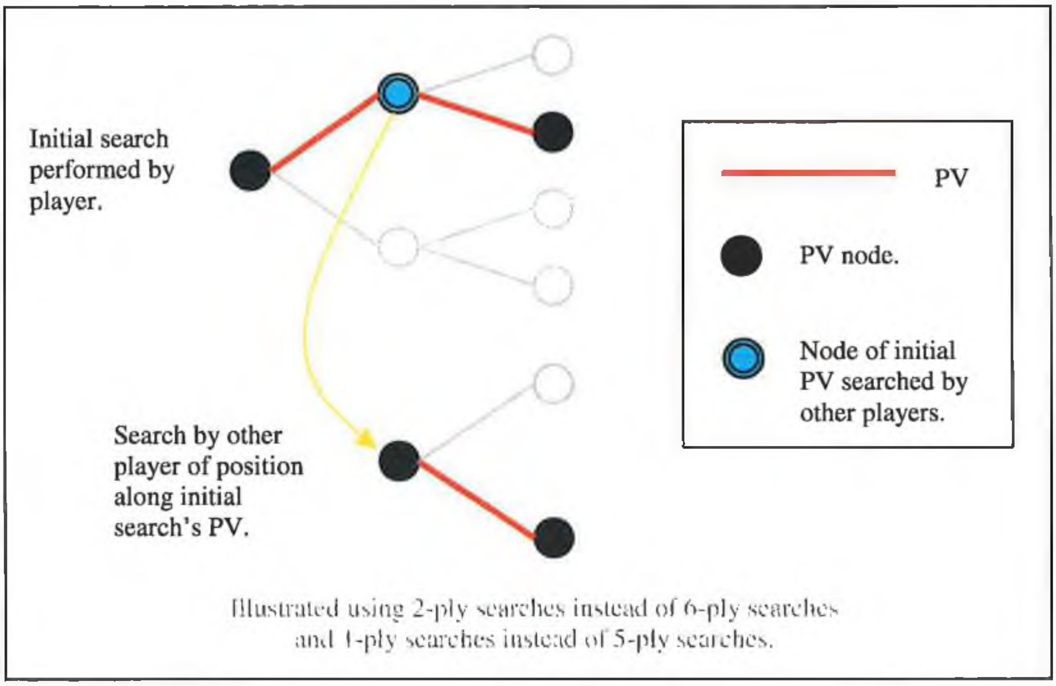



**Figure 6.13:** Diagram illustrating MPSM 2.

Performing tests using the NvNADSP teams set-up using MPSM 1 creates the following tables and results:

| No. | Opening Sequence | Result |
|---|---|---|
| 1 | Centre - 1 | White |
| 2 | Centre - 2 | White |
| 3 | French Defence - 1 | White |
| 4 | French Defence - 2 | White |
| 5 | Kings Indian Defence - 1 | Draw |
| 6 | Kings Indian Defence - 2 | Draw |
| 7 | Nimzo Indian Defence - 1 | Draw |
| 8 | Nimzo Indian Defence - 2 | White |
| 9 | Queens Gambit Accepted - 1 | White |
| 10 | Queens Gambit Accepted - 2 | White |
| 11 | Queens Gambit Declined - 1 | Draw |
| 12 | Queens Gambit Declined - 2 | Draw |
| 13 | Queens Indian Defence - 1 | Draw |
| 14 | Queens Indian Defence - 2 | Draw |
| 15 | Queens Pawn Games - 1 | Draw |
| 16 | Queens Pawn Games - 2 | Draw |
| 17 | Reti Opening - 1 | Draw |
| 18 | Reti Opening - 2 | White |
| 19 | Ruy Lopez - 1 | White |
| 20 | Ruy Lopez - 2 | White |
| 21 | Scotch - 1 | Black |
| 22 | Scotch - 2 | White |
| 23 | Sicilian Defence - 1 | White |
| 24 | Sicilian Defence - 2 | Draw |

| Result | Result Occurrence | Points per Result | Result Points |
|---|---|---|---|
| White Win | 12 | 0 | 0 |
| Black Win | 1 | 1 | 1 |
| Draw | 11 | 0.5 | 5.5 |
| | | **Total** | **6.5** |

**Test Setup 2:** Game results and scoring for MPSM 2.

Analysis of the merits of each of the MPSMs is very difficult based on their games. Comparing the results of the different game opening sequences is not very informative. The games themselves can often be over 40 moves long and any attempt to analysis the moves of the games would prove to be very difficult and I feel would not be very informative either. Therefore comparisons of the MPSMs will be based solely on the results that they achieve.

## 6.7 Problem of Draws and Endgames with Test System and Solution

*Problem with Draws*

In the game of chess a game is declared drawn if a side makes the same move on the same board position three times. A problem in the test system is that the player agents, the entities that are performing the actual searches of the game tree, and the team agents do not have the ability to recognise when a game will be declared drawn because of move repetition. This could result in a game being declared drawn even when a team has a distinct advantage of winning.

In grandmaster matches since 1991 [Opening's Statistics URL] the average percentage of draws from the test set opening sequences is 29%. The percentage of draws with MPSM 2 in Test Setup 2 was 46%; almost half the games were drawn. Some of these draws were in positions that were not ‘natural’ draws, a side had an advantage that it could have exploited to win the game or the drawn game was still in an early stage when draws would not normally occur. The reason for these draws was the test system’s mishandling of the previous states of the game. This is a problem as draw results may be assigned to games despite a side's domination of the game and therefore the strength of the sides are maybe not being properly represented in the test results.

*Problem with Endgames*

A possible factor for the low number of wins for the multiple player team may be sub-optimal play during the endgame since certain personalities, and in particular the Aggressive personality, are not suited to playing endgames. If this were so then the multiple player team would always have a distinct disadvantage of winning even if ahead before entering the endgame. If games were halted when they entered endgame (endgame could possibly be determined by material value on board) the 'natural' result of the game could be reached by playing the rest of the game with a third party chess program.

*Solution*

To resolve games that may have been declared drawn because of the inability of players to recognise repeated moves, the remainder of games are played using GNU Chess 5 [GNU Chess URL, 2001]. The positions for GNU Chess to play are the last positions that were evaluated by white and black. In this way neither of the teams should receive an advantage from getting first move in GNU Chess. The program plays these board positions as machine versus machine with 5 minutes on the clock each. Whatever the result of this play, that is the result assigned to the game. The numeral value given to the games are given in Table 6.14 (values are in terms of black, the multiple player team).

| White Moves 1st | Black Moves 1st | Result |
|---|---|---|
| | | |
| B | B | 1 |
| B | D | .75 |
| B | W | .5 |
| | | |
| D | B | .75 |
| D | D | .5 |
| D | W | .25 |
| | | |
| W | B | .5 |
| W | D | .25 |
| W | W | 0 |

*(B = Black, D = Draw and W = White)*

**Table 6.14:** Results assigned to games using GNU Chess for drawn games and endgames.

A similar arrangement can be used to resolve the problem of sub-optimal endgame play by the multiple player team. Before this can be used, however, it must first be decided when a game can be considered to be in an endgame state. The method used for testing is that used by the Crafty chess program. A game is considered in Crafty to be in an endgame state when the material value of both teams is less than 17, given that material values are Pawn = 0, Knight =3, Bishop = 3, Rook = 5, Queen = 9 and King = 0.

## 6.8 Re-testing using Solution to Draws and Endgames Problem

The term "END-GAME" will be used to indicate when the solution to resolve draws and endgames is being used. Tests were conducted with various MPSMs with and without "END-GAME" running. Results of these tests are given in Table 6.15.

| Test Setup | MPSM | END-GAME | Points |
|---|---|---|---|
| | | | |
| 1 | 1 | ----- | 7 |
| 3 | 1 | END-GAME | 6 |
| | | | |
| 2 | 2 | ----- | 6.5 |
| 4 | 2 | END-GAME | 8.25 |
| | | | |
| 5 | 3* | ----- | 6 |
| 6 | 3* | END-GAME | 9.5 |

**MPSM 3 will be described in Section 6.9*

**Table 6.15:** Tests of MPSMs with and without "END-GAME" running.

The results for the MPSM 2 and MPSM 3 test setups indicate that the multiple player team performs better with "END-GAME" running. MPSM 1 suffers only slightly with "END-GAME" running. The results suggest that the multiple player team is at a disadvantage without "END-GAME" running due to repetition of moves and poor endgame play. Because of this result "END-GAME" will be used for the tests that follow.

## 6.9 Description of MPSM 3

MPSM 3 differs from MPSM 2 in how it assigns values to the evaluations by players of the PVs. MPSM 2 and MPSM 1 ordered the PVs on their evaluations and gave the PV a value based on where they were ordered (see Table 6.16).

| **Evaluation** | +43 | +23 | -129 | +6 | -78 |
|---|---|---|---|---|---|
| **Order** | 1st | 2nd | 5th | 3rd | 4th |
| **Value** | 0 | 1 | 4 | 2 | 3 |

**Table 6.16:** Calculation of PV scores using MPSM 2.

MPSM 3 differs by giving the value based on where the evaluation lies between the best evaluation, given value 0, and the worst evaluation, given value 1.

| **Evaluation** | +43 | +23 | -129 | +6 | -78 |
|---|---|---|---|---|---|
| **Value** | 1 | 0.8837 | 0 | 0.7848 | 0.2965 |

**Table 6.17:** Calculation of PV scores using MPSM 3.

As before the PV with the lowest sum of its values is used as the team solution.

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | W | 0 |
| 2 | Centre - 2 | White | | | 0 |
| 3 | French Defence - 1 | END-GAME | W | W | 0 |
| 4 | French Defence - 2 | END-GAME | W | W | 0 |
| 5 | Kings Indian Defence - 1 | END-GAME | W | W | 0 |
| 6 | Kings Indian Defence - 2 | END-GAME | B | B | 1 |
| 7 | Nimzo Indian Defence - 1 | Black | | | 1 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | D | D | 0.5 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | | | |
| 10 | Queens Gambit Accepted - 2 | END-GAME | B | W | 0.5 |
| 11 | Queens Gambit Declined - 1 | END-GAME | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | D | W | 0.75 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | END-GAME | D | W | 0.25 |
| 15 | Queens Pawn Games - 1 | END-GAME | D | D | 0.5 |
| 16 | Queens Pawn Games - 2 | END-GAME | D | D | 0.5 |
| 17 | Reti Opening - 1 | END-GAME | B | B | 1 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | END-GAME | W | W | 0 |
| 20 | Ruy Lopez - 2 | END-GAME | W | W | 0 |
| 21 | Scotch - 1 | Black | | | 1 |
| 22 | Scotch - 2 | END-GAME | W | W | 0 |
| 23 | Sicilian Defence - 1 | END-GAME | W | W | 0 |
| 24 | Sicilian Defence - 2 | END-GAME | W | B | 0.5 |
| | | | | **Total** | **9.5** |

**Test Setup 6:** MPSM 3, "END-GAME"

| Test Setup | MPSM | END-GAME | Points |
|---|---|---|---|
| 3 | 1 | END-GAME | 6 |
| 4 | 2 | END-GAME | 8.25 |
| 6 | 3 | END-GAME | 9.5 |

**Table 6.18:** Tests of MPSMs with "END-GAME" running.

Quantifying the players' PV preferences linearly rather than by ordering gives a better representation of players' PV preferences. This fact is reflected by the increase in points from MPSM 2 to MPSM 3 (see Table 6.18).

## 6.10 Description of MPSM 4

MPSM 4 is the same as MPSM 3 except in one respect. In MPSM 3 each player performs one initial 6-ply search, their PV answer, and then a 5-ply search for each of the resultant boards from each players' PV initial move. In MPSM 4 these 5-ply searches are replaces by 6-ply searches.

Therefore, in MPSM 4 the multiple player team will search to a depth of 7-ply but only in respect to at most 5 different initial moves.

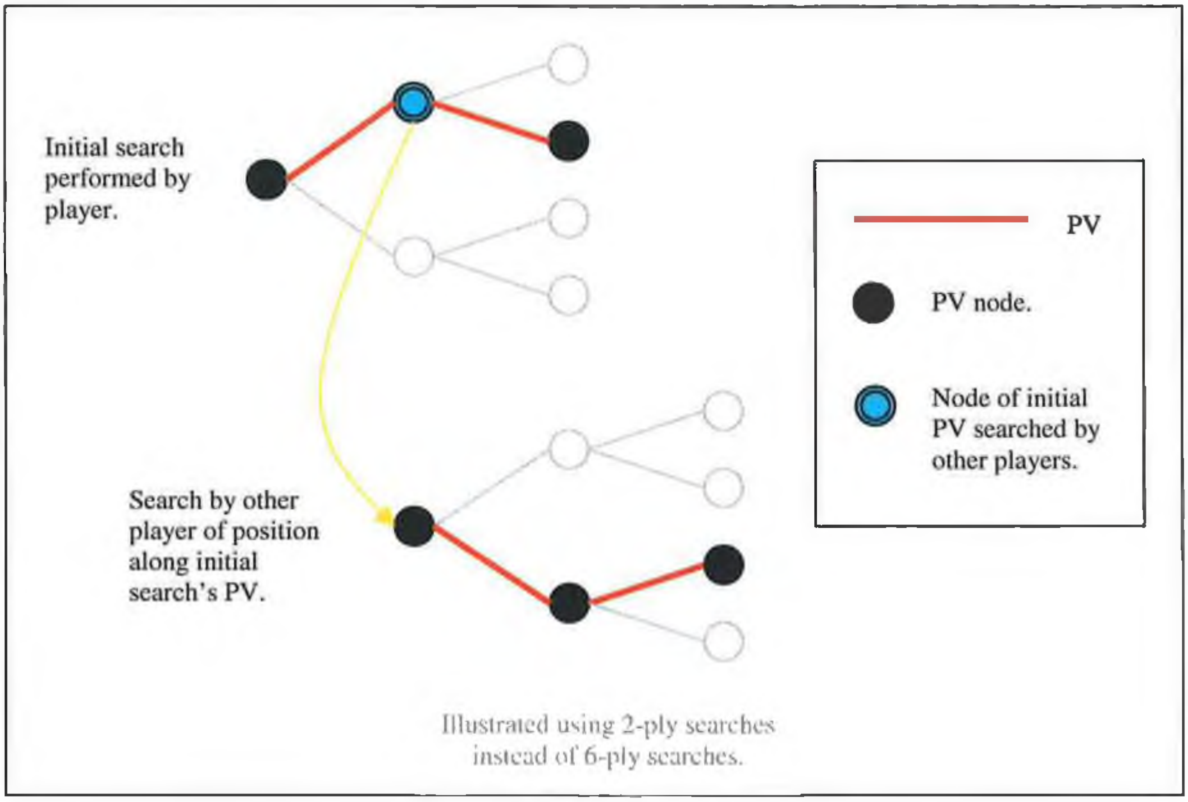



**Figure 6.19:** Diagram illustrating MPSM 4.

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | W | 0 |
| 2 | Centre - 2 | White | | | 0 |
| 3 | French Defence - 1 | Draw | B | B | 1 |
| 4 | French Defence - 2 | END-GAME | D | D | 0.5 |
| 5 | Kings Indian Defence - 1 | Draw | W | W | 0 |
| 6 | Kings Indian Defence - 2 | Draw | D | D | 0.5 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | B | B | 1 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | B | D | 0.75 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | W | W | 0 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | W | W | 0 |
| 11 | Queens Gambit Declined - 1 | Black | | | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | B | B | 1 |
| 13 | Queens Indian Defence - 1 | Draw | D | W | 0.25 |
| 14 | Queens Indian Defence - 2 | Draw | | | |
| 15 | Queens Pawn Games - 1 | Draw | B | D | 0.75 |
| 16 | Queens Pawn Games - 2 | END-GAME | W | W | 0 |
| 17 | Reti Opening - 1 | Draw | W | W | 0 |
| 18 | Reti Opening - 2 | END-GAME | D | D | 0.5 |
| 19 | Ruy Lopez - 1 | END-GAME | B | B | 1 |
| 20 | Ruy Lopez - 2 | Draw | W | W | 0 |
| 21 | Scotch - 1 | END-GAME | B | B | 1 |
| 22 | Scotch - 2 | END-GAME | W | B | 0.5 |
| 23 | Sicilian Defence - 1 | END-GAME | B | B | 1 |
| 24 | Sicilian Defence - 2 | END-GAME | W | W | 0 |
| | | | | Total | 10.75 |

**Test Setup 7:** MPSM 4, "END-GAME"

| Test Setup | MPSM | END-GAME | Points |
|---|---|---|---|
| 3 | 1 | END-GAME | 6 |
| 4 | 2 | END-GAME | 8.25 |
| 6 | 3 | END-GAME | 9.5 |
| 7 | 4 | END-GAME | 10.75 |

**Table 6.20:** Tests of MPSMs with "END-GAME" running.

MPSM 4 results in a further improvement in the test system. As can be seen in Table 6.20, each improved MPSM has increased the number of points achieved by the multiple player team. There are, however, problems with MPSM 4 that will be discussed next.

## 6.11 Horizon Effect en Masse Problem with MPSM 4

MPSM 4 can suffer from the horizon effect *en masse*. This is demonstrated in the Figure 6.21 involving a team consisting of two players using 2-ply searches instead of 6-ply searches.

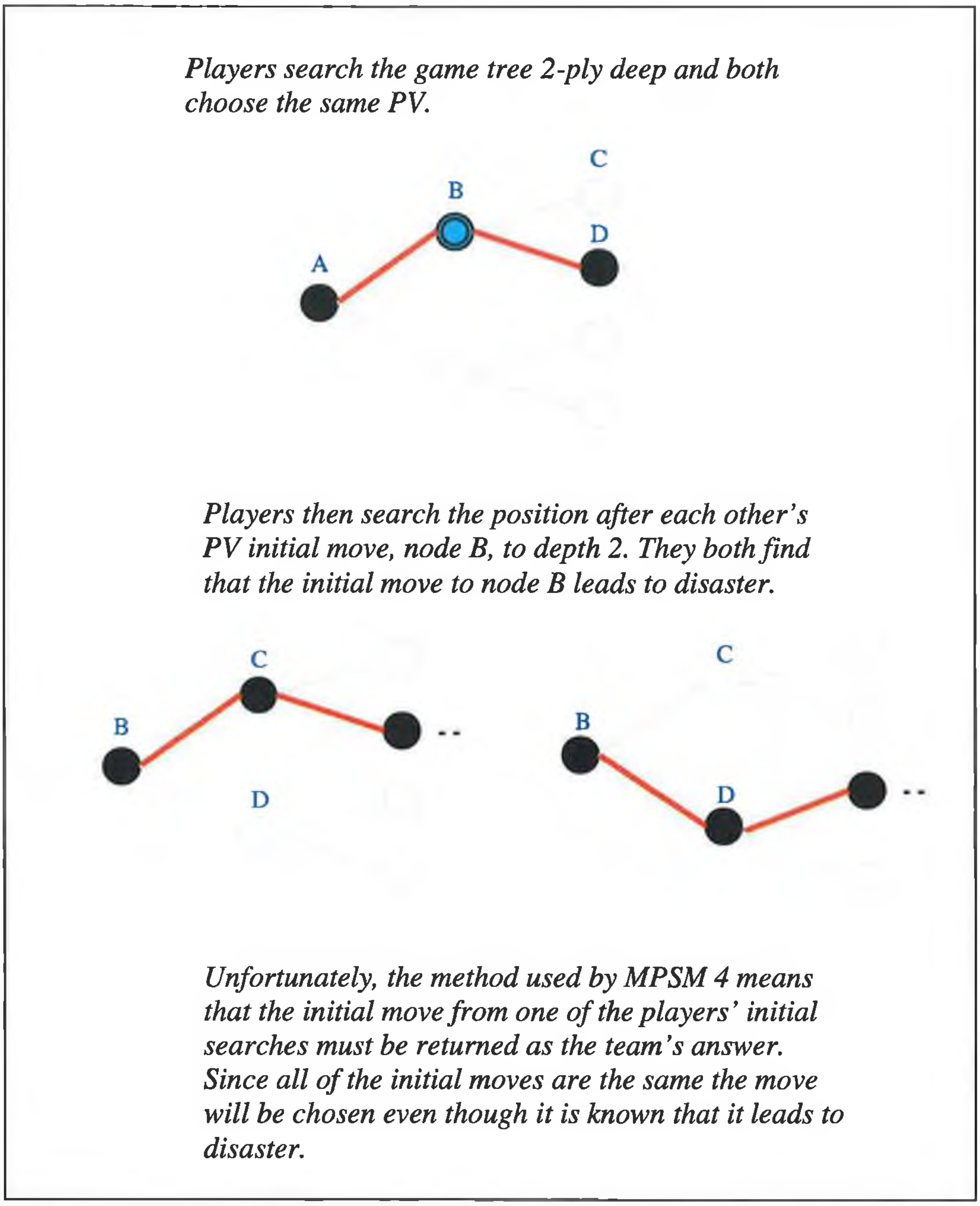



**Figure 6.21:** Demonstrates horizon effect of MPSM 4 using 2-ply searches.

Table 6.22 shows the frequency of different initial moves among the player PVs for a game using MPSM 4. In the game, the MPSM only has at most two initial moves to choose from 64% of the time. This illustrates the problem of horizon effect *en masse* in MPSM 4.

| Number of Different Initial Moves in Players' PVs | Frequency |
|---|---|
| 5 | 0.0 % |
| 4 | 12.5 % |
| 3 | 21.875 % |
| 2 | 43.75 % |
| 1 | 21.875 % |

**Table 6.22:** Frequency of number of different initial PV moves in game using MPSM 4.

## 6.12 Solution to Horizon Effect en Masse

There are two possible methods of resolving this problem. They are:

1) To replace the PVs with duplicated initial moves with other initial moves instead. This would involve finding all duplicates and replacing them with good moves. The question is what moves would take the place of a duplicated move? To find these moves would further searches need to be performed?
2) Implement quiescense search (see Section 2.3.4) into the test system. Instead of calling the evaluation function when depth is zero, quiescence search would be called. Quiescence search continues the

search of the game tree but only expands a position for capture and promotion moves. The purpose of quiescence search is to push the effects of horizon effect further away and hence decrease its influence on test results.

Quiescence search was chosen as it is present in most chess programs and so least affects the practicality of the tests. The other reason is the ad-hoc nature of the first proposed method. The method of replacing duplicated moves is unclear and would obscure the ideas behind this research.

Due to time constraints quiescence search was limited to a search depth of 6-ply. Both teams implement the 6-ply quiescence search so any gain by the multiple player team is not due to any implementation advantage.

### 6.13 Tests with Quiescence Search

Tables 6.23 and 6.24 give the results of tests with and without quiescence search respectively. For all the MPSMs the multiple player team achieved a better result with quiescence search running. This suggests that the addition of quiescence search into the test system prevents the multiple player team from suffering as badly from the horizon effect *en masse* as described previously.

| Test Setup | MPSM | END-GAME | Quiescence | Points |
|---|---|---|---|---|
| 3 | 1 | END-GAME | No | 6 |
| 4 | 2 | END-GAME | No | 8.25 |
| 6 | 3 | END-GAME | No | 9.5 |
| 7 | 4 | END-GAME | No | 10.75 |

**Table 6.23:** MPSM tests without quiescence search running.

| Test Setup | MPSM | END-GAME | Quiescence | Points |
|---|---|---|---|---|
| 8 | 1 | END-GAME | Yes | 10.75 |
| 9 | 2 | END-GAME | Yes | 11 |
| 10 | 3 | END-GAME | Yes | 15.25 |
| 11 | 4 | END-GAME | Yes | 15.75 |

**Table 6.24:** MPSM tests with quiescence search running.

## 6.14 Performance of Black as Single Player Team

The games played during testing have been of the form NvNADSP, a single player team versus a multiple player team. The multiple player team is taking the place of what would usually be a single player team in current chess programs, the games of which would be of the form NvN.

So far the performance of the multiple player team has been rated in terms of its opposition to the single player team, who is white. A better way to gauge the progress of the multiple player team is to compare its results against the results of a single player team, similar to the white team, in its place. This would demonstrate the possible superiority of the multiple player team to the traditional single player team.

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | W | 0 |
| 2 | Centre - 2 | END-GAME | W | W | 0 |
| 3 | French Defence - 1 | END-GAME | W | W | 0 |
| 4 | French Defence - 2 | Draw | W | W | 0 |
| 5 | Kings Indian Defence - 1 | Black | | | 1 |
| 6 | Kings Indian Defence - 2 | Draw | W | W | 0 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | W | W | 0 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | B | B | 1 |
| 9 | Queens Gambit Accepted - 1 | Black | | | 1 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | W | W | 0 |
| 11 | Queens Gambit Declined - 1 | END-GAME | W | W | 0 |
| 12 | Queens Gambit Declined - 2 | Draw | B | B | 0 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | Black | | | 1 |
| 15 | Queens Pawn Games - 1 | White | | | 0 |
| 16 | Queens Pawn Games - 2 | END-GAME | B | B | 1 |
| 17 | Reti Opening - 1 | White | | | 0 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | END-GAME | B | B | 1 |
| 20 | Ruy Lopez - 2 | END-GAME | W | W | 0 |
| 21 | Scotch - 1 | END-GAME | W | W | 0 |
| 22 | Scotch - 2 | Draw | D | D | 0.5 |
| 23 | Sicilian Defence - 1 | Black | | | 1 |
| 24 | Sicilian Defence - 2 | END-GAME | W | W | 0 |
| | | | | **Total** | **8.5** |

**Test Setup 12:** NvN, "END-GAME", No Quiescence Search Running.

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | Draw | D | D | 0.5 |
| 2 | Centre - 2 | END-GAME | B | D | 0.75 |
| 3 | French Defence - 1 | END-GAME | B | B | 1 |
| 4 | French Defence - 2 | END-GAME | D | D | 0.5 |
| 5 | Kings Indian Defence - 1 | END-GAME | W | W | 0 |
| 6 | Kings Indian Defence - 2 | END-GAME | B | B | 1 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | W | B | 0.5 |
| 8 | Nimzo Indian Defence - 2 | Draw | B | B | 1 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | B | B | 1 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | D | B | 0.75 |
| 11 | Queens Gambit Declined - 1 | Draw | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | B | B | 1 |
| 13 | Queens Indian Defence - 1 | Draw | W | W | 0 |
| 14 | Queens Indian Defence - 2 | END-GAME | D | D | 0.5 |
| 15 | Queens Pawn Games - 1 | END-GAME | D | D | 0.5 |
| 16 | Queens Pawn Games - 2 | END-GAME | B | W | 0.5 |
| 17 | Reti Opening - 1 | END-GAME | B | B | 1 |
| 18 | Reti Opening - 2 | END-GAME | B | B | 1 |

| | | | | | |
|---|---|---|---|---|---|
| **19** | Ruy Lopez - 1 | END-GAME | W | W | 0 |
| **20** | Ruy Lopez - 2 | Draw | W | W | 0 |
| **21** | Scotch - 1 | END-GAME | B | B | 1 |
| **22** | Scotch - 2 | END-GAME | B | B | 1 |
| **23** | Sicilian Defence - 1 | Draw | B | B | 1 |
| **24** | Sicilian Defence - 2 | END-GAME | D | D | 0.5 |
| | | | | **Total** | **16** |

**Test Setup 13:** NvN, "END-GAME", Quiescence Search Running.

As discussed in Section 6.3, it was expected that white would have a small advantage over black because of getting to move first. This is confirmed by Test Setup 12 in which the black team achieves a result of 8.5, a result of 12 would indicate that both teams are evenly matched. However, the result of Test Setup 13 does not conform to this pattern in which black achieves a result of 16, twice that achieved by white. This result does not indicate an error in the test system but rather the limitations imposed by the small test data set (24 games per test setup). These limitations will be discussed further in the Section 7 of this work.

## 6.15 Analysis

The purpose of the research is to investigate the possible use of multiple personalities in the game of chess. The collaboration of the personalities is controlled by the MPSMs and development of these is central to the research. Tests were performed using various MPSMs with and without quiescence search running. From the tests, Tables 6.25 and 6.26 can be extracted.

| Test Setup | MPSM | END-GAME | Quiescence | Points |
|---|---|---|---|---|
| 3 | 1 | END-GAME | No | 6 |
| 4 | 2 | END-GAME | No | 8.25 |
| 6 | 3 | END-GAME | No | 9.5 |
| 7 | 4 | END-GAME | No | 10.75 |
| 12 | NvN | END-GAME | No | 8.5 |

**Table 6.25:** List of tests without quiescence search running.

| Test Setup | MPSM | END-GAME | Quiescence | Points |
|---|---|---|---|---|
| 8 | 1 | END-GAME | Yes | 10.75 |
| 9 | 2 | END-GAME | Yes | 11 |
| 10 | 3 | END-GAME | Yes | 15.25 |
| 11 | 4 | END-GAME | Yes | 15.75 |
| 13 | NvN | END-GAME | Yes | 16 |

**Table 6.26:** List of tests with quiescence search running.

The purpose of the multiple player team is to achieve a better result than a single player team in the same position. These single player black teams are indicated by the NvN tests. The black side in the NvN games perform 7-ply searches just like the single player white sides. Therefore comparing the NvNADSP and NvN results is similar to comparing the strength of the multiple player teams, with their 6-ply searches, against a single player team with 7-ply searches.

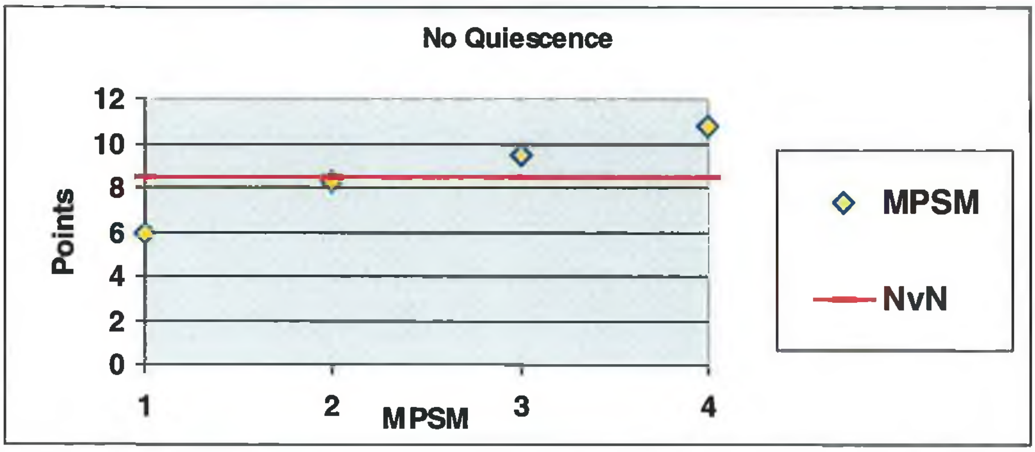



**Figure 6.27:** Chart of tests without quiescence search.

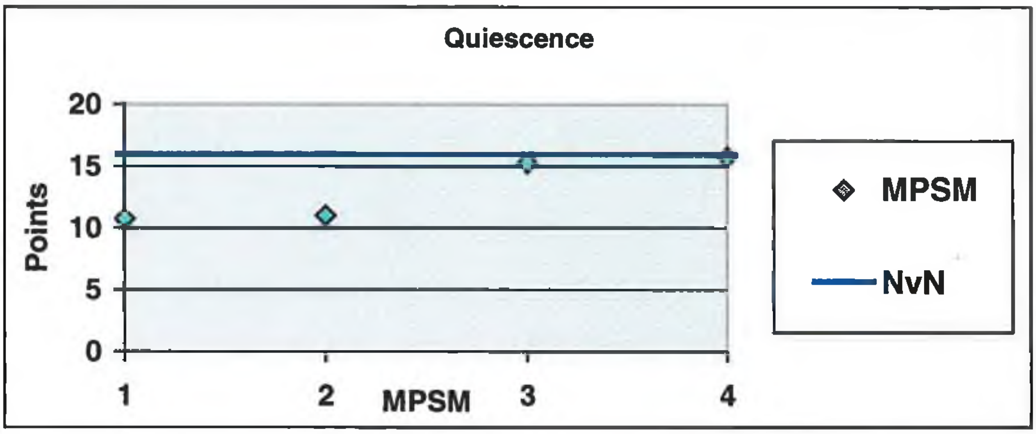



**Figure 6.28:** Chart of tests with quiescence search.

As can be seen in Figures 6.27 and 6.28 above, each MPSM was an improvement on its descendants. The aim for the MPSMs is to achieve better results than their NvN counterparts. In the tests without quiescence running (Figure 6.27) this aim is achieved by MPSM 3 and MPSM 4. In tests with quiescence running (Figure 6.28) MPSM 4 almost make it to the NvN level.

The tests suggest that multiple personality teams may be a practical alternative to the single evaluation function architecture of modern chess programs but the tests do contain many limitations. The small size of the test set, 24 opening board sequences, imposes limits upon the significance that can be gleaned from the results. Also, the introduction of GNU Chess 5 to resolve draws and endgames adds further complexity to the tests and weakens the relationship between the multiple player teams and their results.

# Chapter 7

# Conclusions

Computer chess and the advent of modern computers occurred almost simultaneously. Since then computer chess has played a pivotal role in AI research and has been the test-bed for many of the advances in AI. Although progress was often considered to be slow computer chess has progressed immensely since Alan Turing wrote the first chess program in 1950. This progress culminated with Deep Blue's famous victory over Garry Kasparov, probably the strongest chess player who ever lived, in 1997. This victory demonstrated that computers could achieve excellence in a domain that was considered "a touchstone of the intellect" by Goethe.

The victory by Deep Blue, however, needs to be put into perspective against the other AI problems that need to be resolved. Deep Blue was not a computer in the normal sense, which could perform multiple different tasks, but rather a purpose built machine to play chess using specially constructed hardware. The hardware required to perform this feat is far beyond what is publicly available. Chess, with a branching factor of around 36-40, is a simple game compared to more complex games such as the Oriental game Go, which has a branching factor of 180. The victory by Deep Blue demonstrated not only the possible power of computers but also the progress required if games such as Go are to be solved.

## 7.1 Purpose of this Research

The inspiration for the research presented in this thesis was the Kasparov versus the World match that began on June 21st 1999 over the Internet. The contest involved a single game of chess between Garry Kasparov and a world team consisting of players from all around the world. The collaborating world team players strongly challenged Kasparov for the match but soon fell off the rails as allegations of vote stuffing and incompetence by match organisers arose. The match concluded on the 23rd October 1999 with victory to Kasparov. Despite the controversies that marred the match, the contest demonstrated the power of collaboration in chess with the World team achieving better play than any of the team members could have achieved alone.

Due to the complexity of human chess collaboration, a simplified version of chess collaboration was investigated. The World team consisted of players with different styles of play collaborating. Style of play in computer chess was found to be located within the evaluation function of a program. By creating multiple evaluation functions, with different payoffs for board features, multiple 'personalities' with different styles of play were created. The purpose of this research was to create different collaborating methods to find if these personalities could collaborate to create better chess play.

## 7.2 Summary of Results

Tests were performed using multiple player teams comprising of 5 players, with search depth of 6-ply, and single player teams, with search depth of 7-ply. A test set of 24 different starting positions, taken from master chess opening play, was decided upon. Separate tests were conducted with and without quiescence search of 6-ply enabled.

Multiple Player Solution Methods (MPSMs) were developed to enable the personalities in the multiple player teams to collaborate. Four different MPSMs were created as testing was conducted, each one building upon the methods employed by its predecessors. Tests took the form of NvNADSP, each letter indicating the players involved in each team. The multiple player team (NADSP), which uses the MPSMs, took the role of black in the tests. The black side in the game of chess is at a small disadvantage by not going first. To remedy this and to get direct comparisons between single player teams and multiple player teams, tests were performed to measure the level of performance by a single player team in the same initial position (NvN).

As illustrated in the summary of results in Section 6.15, each improved MPSM resulted in better chess play both with and without quiescence search. In tests without quiescence search enabled, NvNADSP outperformed NvN by MPSM 3 and continued its progress with MPSM 4. In tests with quiescence search enabled, NvNADSP with MPSM 4 almost achieved the same level of performance as NvN. Each MPSM resulted in an improved performance over its predecessor suggesting that further development of the MPSM could enable the multiple player team to outperform the single player team, in tests with quiescence search enabled. These results suggest that a chess program with multiple styles of play collaborating could be a practical alternative to the single style of play paradigm currently used by computer chess programs.

Despite the attractiveness of taking the conclusions of the results as a true reflection of the quality of using multiple collaborating personalities, the tests had a number of drawbacks. Tests were conducted using a test set of 24 different positions from master chess opening play. This data set is too small to conclude conclusively the merits of the results. A data set of significantly more starting positions would be required to get results that could properly indicate the true value of using multiple collaborative personalities. Unfortunately due to time constraints such a test data set was not possible for this work. Also, the test system created contained a flaw that hindered the team and player agents from being able to tell when a game would be declared a draw because of move repetition. This was resolved using a third party chess program to assign a more accurate result to the game. This, however, added further complexity to the results may have weakened the validity of the results.

### 7.3 Future Work

Future work in this area would principally be concerned with the implementation of collaborative multiple personalities, as presented in this work, into a strong chess program. Computer chess programming is a very long and difficult task and one could possibly forgo much of this work by using code from an existing chess program. GNU Chess and Crafty are two strong computer chess programs whose source code is publicly available. Whether or not these programs are suitable for modification to allow collaborative multiple personalities is unknown at this time.

Implementing the idea of multiple personalities into a single threaded program, such as GNU Chess or Crafty, has many advantages over the multiple threaded approach taken in this work. In the multiple threaded approach, the amount of processing given to a side was controlled using search depths. By using a single threaded approach search could use time control, allowing the program to compete in the same arena as other chess programs and hence its playing strength could be easily measured. By using a single threaded program, branches of the game tree would not need to be recreated for each personality as in the multiple threaded approach. The search though the game tree could maintain a list of the personalities and their current state data, such as their alpha and beta values, allowing just one version of the game tree to be created. Cut-offs would be possible once all personalities had indicated that the branch is of no consequence.

# References

[Beal 1990] D. F. Beal. A generalized quiescence search algorithm. *Artificial Intelligence, 43(1):85-- 98, 1990.*

[Carmel and Markovitch 1994] David Carmel and Shaul Markovitch. The M* Algorithm: Incorporating Opponent Models into Adversary Search. *Technical Report CIS9402, Israel Institute of Technology, Computer Science Department, March 1994.*

[Chase and Simon 1973] W.G. Chase and H.A. Simon. Perception in Chess. *Cognitive Psychology, 4:55-81, 1973.*

[Chess Corner URL] Chess Corner website, World Champions Section. http://www.chesscorner.com/worldchamps/karpov/karpov.htm (Last visited – July 2002)

[ChessLab URL] ChessLab website. Chess games online database search. http://www.chesslab.com/PositionSearch.html (Last visited – July 2002)

[CORBA URL] Object Management Group's CORBA Website. http://www.corba.org/ (Last visited – July 2002)

[Crafty URL] Robert Hyatt's homepage, http://www.cis.uab.edu/info/faculty/hyatt/hyatt.html (Last visited – July 2002)

[Davis 1997] Morton D. Davis. Game Theory: A Non-Technical Introduction. *Dover Publications, 1997.*

[de Groot 1965] A. D. de Groot. Thought and choice in chess. *The Hague: Mouton, 1965.*

[Deep Blue URL] Deep Blue homepage.
http://www.research.ibm.com/deepblue/home/html/b.html
(Last visited – July 2002)

[Elo 1978] Arpad Elo. The Rating of Chessplayers, Past and Present. *Arco Publishing, New York, 1978.*

[Feldmann 1997] Rainer Feldmann. Computer Chess: Algorithms and Heuristics for a Deep Look into the Future. *Conference on Current Trends in Theory and Practice of Informatics, 1997.*

[Flinter and Keane 1995] Flinter, S. and M. T. Keane. Using chunking for the automatic generation of cases in chess. *Proceedings of the 1st International Conference on Case Based Reasoning, 1995.*

[Franklin and Graesser 1996] Stan Franklin and Art Graesser. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.*

[GNU Chess URL, 2001] GNU Chess URL, http://www.gnu.org/software/chess/
(Last visited – July 2002)

[Heinz 1997] Ernst A. Heinz. How DarkThought Plays Chess. *ICCA Journal, Vol. 20(3), pages 166-176, September 1997.*

[IBM Tokyo URL] IBM Tokyo Research Laboratory, Aglets.
http://www.trl.ibm.com/aglets/index_e.htm (Last visited – July 2002)

[Junghanns et al. 1997] Junghanns A., Schaeffer J., Brockington M., Bjornsson Y. and Marsland T. Diminishing Returns for Additional Search in Chess. *Advances in Computer Chess 8, pages 53-67, 1997.*

[Knuth and Moore 1975] D.E. Knuth and R.W. Moore. An analysis of alpha-beta pruning. *Artificial Intelligence 6,4, pages 293-326, 1975.*

[Korf 1985] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence, vol. 27, no 1, pages 97-109, 1985.*
[Korn 1972] Walter Korn. Modern Chess Openings ($11^{th}$ ed.). *Pitman 1972.*

[Lange and Oshima 1998] Danny B. Lange and Mitsuru Oshima. Programming and Deploying Java Mobile Agents with Aglets. *Addison Wesley, 1998.*

[Levy and Newborn 1991] David Levy and Monty Newborn. How Computers Play Chess, *pages 153-190. Computer Science Press, 1991.*

[López-Ortiz 1993] Alejandro López-Ortiz. Computer Chess: Past to Present. *Expository Paper, University of Waterloo Imprint, 1993.*
http://www.cs.unb.ca/~alopez-o/divulge/chimp.html (Last visited – July 2002)

[Luce and Raiffa 1957] Duncan Luce and Howard Raiffa. Games and Decisions. *Wiley, 1957.*

[McConnell 1995] Chris McConnell. Tuning evaluation functions for search. *Proceedings of the Hong Kong Computer Strategy Game Programming Workshop, 1995.*

[Michie 1982] Donald Michie. Machine Intelligence and Related Topics. *Gordon and Breach Science Publishers, 1982.*

[Microsoft PressPass URL] Microsoft Press Release for Kasparov versus the World competition.
http://www.microsoft.com/presspass/press/1999/Jun99/ChessPR.ASP
(Last visited – July 2002)

[MSN Kasparov URL] Kasparov versus the World website.
http://www.zone.com/kasparov/Home.asp (Last visited – July 2002)

[Opening's Statistics URL] Chess Opening's Statistics http://homepage.ntlworld.com/adam.bozon/stats.htm (Last visited – July 2002)

[Orfali et al. 1999] Robert Orfali, Dan Harkey and Jeri Edwards. Client/Server Survival Guide 3rd Edition. *John Wiley & Sons, Inc., 1999.*

[Philidor URL] Francois-Andre Danican Philidor biography. http://www.geocities.com/siliconvalley/lab/7378/philidor.htm (Last visited – July 2002)

[Shannon 1950] Claude Shannon. Programming a computer for playing chess. *Philosophical Magazine, Vol. 41, pages 256-275, 1950.*

[Simon and Schaeffer 1992] Herbert Simon and Jonathan Schaeffer. The Game of Chess. *Handbook of Game Theory with Economic Applications, vol. 1, 1992.*

[Sinclair 1998] David Sinclair. Using Example-Based Reasoning for Selective Move Generation in Two Player Adversarial Games. *Proceedings of the 4th European Workshop on Case-Based Reasoning, LNAI, 1998.*

[TSCP URL] Tom Kerrigan's homepage. http://home.attbi.com/~tckerrigan/ (Last visited – July 2002)

[Turing 1950] Alan Turing. Computing Machinery and Intelligence. *Mind, 59:434-460, 1950.*

[Uiterwijk 1995] Jos Uiterwijk. Computer Chess: State of the Art. *Proceedings of the Game Programming Workshop in Japan, 1995.*

[Verhelst URL] Paul Verhelst. Chess Tree Search. http://www.xs4all.nl/~verhelst/chess/search.html (Last visited – July 2002) [von Neumann and Morgenstern 1944] John von Neumann and Oskar Morgenstern. Theory of Games and Economic Behavior. *Princeton University Press, 1944.*

# Appendix A
## FIDE Laws of Chess
*(Abridged)*

**RULES OF PLAY**

**Article 1: The nature and objectives of the game of chess**

1.1 The game of chess is played between two opponents who move their pieces alternately on a square board called a 'chessboard'. The player with the white pieces commences the game. A player is said to 'have the move', when his opponent's move has been made.

1.2 The objective of each player is to place the opponent's king 'under attack' in such a way that the opponent has no legal move which would avoid the 'capture' of the king on the following move. The player who achieves this goal is said to have 'checkmated' the opponent's king and to have won the game. The opponent whose king has been checkmated has lost the game.

1.3 If the position is such that neither player can possibly checkmate, the game is drawn.

**Article 2: The initial position of the pieces on the chessboard**

2.1 The chessboard is composed of an 8x8 grid of 64 equal squares alternately light (the 'white' squares) and dark (the 'black' squares).
The chessboard is placed between the players in such a way that the near corner square to the right of the player is white.

2.2 At the beginning of the game one player has 16 light-coloured pieces (the 'white' pieces); the other has 16 dark-coloured pieces (the 'black' pieces): These pieces are as follows:

A white king, usually indicated by the symbol ♔
A white queen, usually indicated by the symbol ♕
Two white rooks, usually indicated by the symbol ♖
Two white bishops, usually indicated by the symbol ♗
Two white knights, usually indicated by the symbol ♘
Eight white pawns, usually indicated by the symbol ♙
A black king, usually indicated by the symbol ♚
A black queen, usually indicated by the symbol ♛
Two black rooks, usually indicated by the symbol ♜
Two black bishops, usually indicated by the symbol ♝
Two black knights, usually indicated by the symbol ♞
Eight black pawns, usually indicated by the symbol ♟

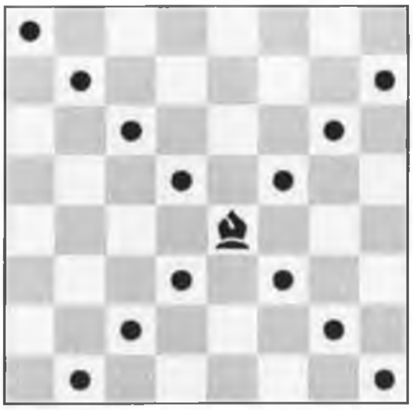
2.3 The initial position of the pieces on the chessboard is as follows:

2.4 The eight vertical columns of squares are called 'files'. The eight horizontal rows of squares are called ranks'. A straight line of squares of the same colour, touching corner to corner, is called a 'diagonal'.

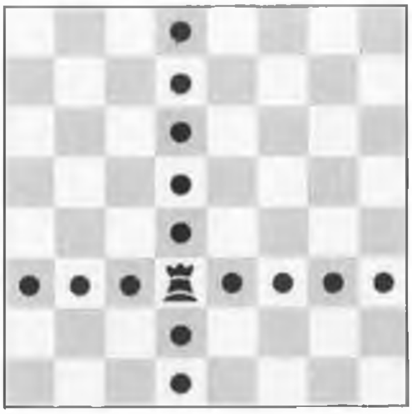
**Article 3: The moves of the pieces**

3.1 It is not permitted to move a piece to a square occupied by a piece of the same colour. If a piece moves to a square occupied by an opponent's piece the latter is captured and removed from the chessboard as part of the same move. A piece is said to attack an opponent's piece if the piece could make a capture on that square according to Articles 3.2 to 3.8.

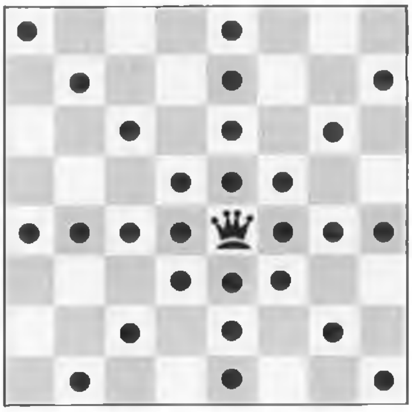
3.2 The bishop may move to any square along a diagonal on which it stands.



3.3 The rook may move to any square along the file or the rank on which it stands.

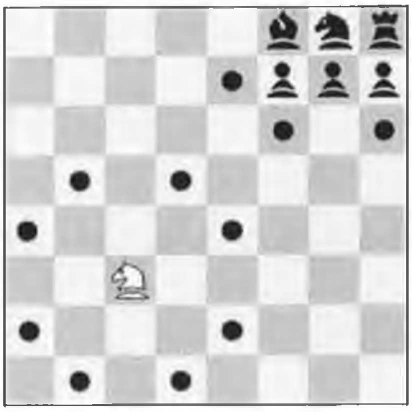


3.4 The queen may move to any square along the file, the rank or a diagonal on which it stands.



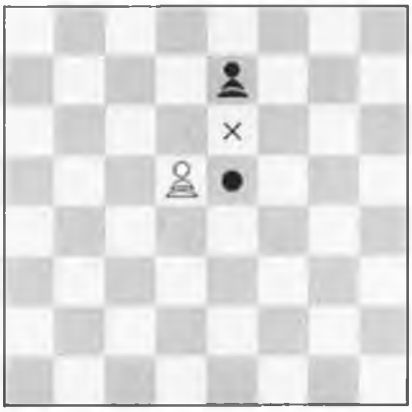
3.5 When making these moves the bishop, rook or queen may not move over any intervening pieces.

3.6 The knight may move to one of the squares nearest to that on which it stands but not on the same rank, file or diagonal.



3.7
a. The pawn may move forward to the unoccupied square immediately in front of it on the same file, or
b. on its first move the pawn may move as in (a); alternatively it may advance two squares along the same file provided both squares are unoccupied, or
c. the pawn may move to a square occupied by an opponent's piece, which is diagonally in front of it on an adjacent file, capturing that piece.

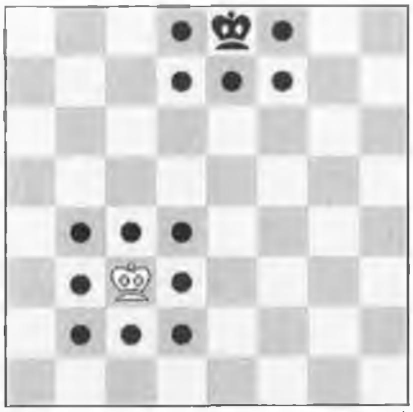
d. A pawn attacking a square crossed by an opponent's pawn which has advanced two squares in one move from its original square may capture this opponent's pawn as though the latter had been moved only one square. This capture may only be made on the move following this advance and is called an 'en passant' capture.



e. When a pawn reaches the rank furthest from its starting position it must be exchanged as part of the same move for a queen, rook, bishop or knight of the same colour. The player's choice is not restricted to pieces that have been captured previously. This exchange of a pawn for another piece is called 'promotion' and the effect of the new piece is immediate.

3.8 a. There are two different ways of moving the king, by:

i. moving to any adjoining square not attacked by one or more of the opponent's pieces.

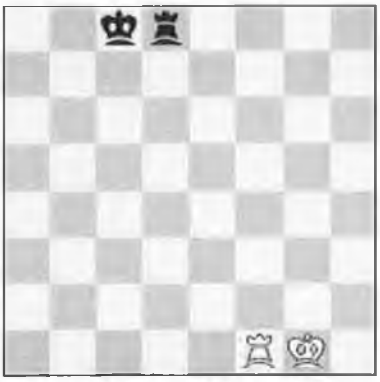


The opponent's pieces are considered to attack a square, even if such pieces cannot themselves move.

or

ii. 'castling'. This is a move of the king and either rook of the same colour on the same rank, counting as a single move of the king and executed as follows: the king is transferred from its original square two squares towards the rook, then that rook is transferred to the square the king has just crossed.

*Before white kingside castling*
*Before black queenside castling*



*After white kingside castling*
*After black queenside castling*

*Before white queenside castling*
*Before black kingside castling*

*After white queenside castling*
*After black kingside castling*

(1) Castling is illegal:
  b. if the king has already moved, or
  b. with a rook that has already moved

(2) Castling is prevented temporarily

b. if the square on which the king stands, or the square which it must cross, or the square which it is to occupy, is attacked by one or more of the opponent's pieces.

b. if there is any piece between the king and the rook with which castling is to be effected.

b. The king is said to be 'in check', if it is attacked by one or more of the opponent's pieces, even if such pieces cannot themselves move. Declaring a check is not obligatory.

3.9 No piece can be moved that will expose its own king to check or leave its own king in check.

**Article 4: The act of moving the pieces**

4.1 Each move must be made with one hand only.

4.2 Provided that he first expresses his intention (e.g. by saying "j'adoube" or "I adjust"), the player having the move may adjust one or more pieces on their squares.

4.3 Except as provided in Article 4.2, if the player having the move deliberately touches on the chessboard

a. one or more of his own pieces, he must move the first piece touched that can be moved, or

b. one or more of his opponent's pieces, he must capture the first piece touched, which can be captured, or

c. one piece of each colour, he must capture the opponent's piece with his piece or, if this is illegal, move or capture the first piece touched which can be moved or captured. If it is unclear, whether the player's own piece or his opponent's was touched first, the player's own piece shall be considered to have been touched before his opponent's.

4.4

a. If a player deliberately touches his king and rook he must castle on that side if it is legal to do so.

b. If a player deliberately touches a rook and then his king he is not allowed to castle on that side on that move and the situation shall be governed by Article 4.3(a).

c. If a player, intending to castle, touches the king or king and rook at the same time, but castling on that side is illegal, the player must make another legal move with his king which may include castling on the other side. If the king has no legal move, the player is free to make any legal move.

4.5 If none of the pieces touched can be moved or captured, the player may make any legal move.

4.6 A player forfeits his right to a claim against his opponent's violation of Article 4.3 or 4.4, once he deliberately touches a piece.

4.7 When, as a legal move or part of a legal move, a piece has been released on a square, it cannot then be moved to another square. The move is considered to have been made when all the relevant requirements of Article 3 have been fulfilled.

**Article 5: The completion of the game**

5.1

a. The game is won by the player who has checkmated his opponent's king. This immediately ends the game, provided that the move producing the checkmate position was a legal move.

b. The game is won by the player whose opponent declares he resigns. This immediately ends the game.

5.2

a. The game is drawn when the player to move has no legal move and his king is not in check. The game is said to end in 'stalemate'. This immediately ends the game, provided that the move producing the stalemate position was legal.

b. The game is drawn when a position has arisen in which neither player can checkmate the opponent's king with any series of legal moves. The game is said to end in a 'dead position'. This immediately ends the game, provided that the move producing the position was legal.

c. The game is drawn upon agreement between the two players during the game. This immediately ends the game. (See Article 9.1)

d. The game may be drawn if any identical position is about to appear or has appeared on the chessboard at least three times. (See Article 9.2)

e. The game may be drawn if each player has made the last 50 consecutive moves without the movement of any pawn and without the capture of any piece. (See Article 9.3)

**Article 9: The drawn game**

9.1

a. A player wishing to offer a draw shall do so after having made a move on the chessboard and before stopping his clock and starting the opponent's clock. An offer at any other time during play is still valid, but Article 12.5 must be considered. No conditions can be attached to the offer. In both cases the offer cannot be withdrawn and remains valid until the opponent accepts it, rejects it orally, rejects it by touching a piece with the intention of moving or capturing it, or the game is concluded in some other way.

b. The offer of a draw shall be noted by each player on his scoresheet with a symbol (See Appendix E).

c. A claim of a draw under 9.2, 9.3 or 10.2 shall be considered to be an offer of a draw.

9.2 The game is drawn, upon a correct claim by the player having the move, when the same position, for at least the third time (not necessarily by sequential repetition of moves)

a. is about to appear, if he first writes his move on his scoresheet and declares to the arbiter his intention to make this move, or

b. has just appeared, and the player claiming the draw has the move.

Positions as in (a) and (b) are considered the same, if the same player has the move, pieces of the same kind and colour occupy the same squares, and the possible moves of all the pieces of both players are the same.
Positions are not the same if a pawn that could have been captured en passant can no longer be captured or if the right to castle has been changed temporarily or permanently.

9.3 The game is drawn, upon a correct claim by the player having the move, if

a. he writes on his scoresheet, and declares to the arbiter his intention to make a move which shall result in the last 50 moves having been made by each player without the movement of any pawn and without the capture of any piece, or

b. the last 50 consecutive moves have been made by each player without the movement of any pawn and without the capture of any piece.

9.4 If the player makes a move without having claimed the draw he loses the right to claim, as in Article 9.2 or 9.3, on that move.

9.5 If a player claims a draw as in Article 9.2 or 9.3, he shall immediately stop both clocks. He is not allowed to withdraw his claim.

a. If the claim is found to be correct the game is immediately drawn.
b. If the claim is found to be incorrect, the arbiter shall add three minutes to the opponent's remaining time. Additionally, if the claimant has more than two minutes on his clock the arbiter shall deduct half of the claimant's remaining time up to a maximum of three minutes. If the claimant has more than one minute, but less than two minutes, his remaining time shall be one minute. If the claimant has less than one minute, the arbiter shall make no adjustment to the claimant's clock. Then the game shall continue and the intended move must be made.

9.6 The game is drawn when a position is reached from which a checkmate cannot occur by any possible series of legal moves, even with the most unskilled play. This immediately ends the game.

**Article 11: Scoring**

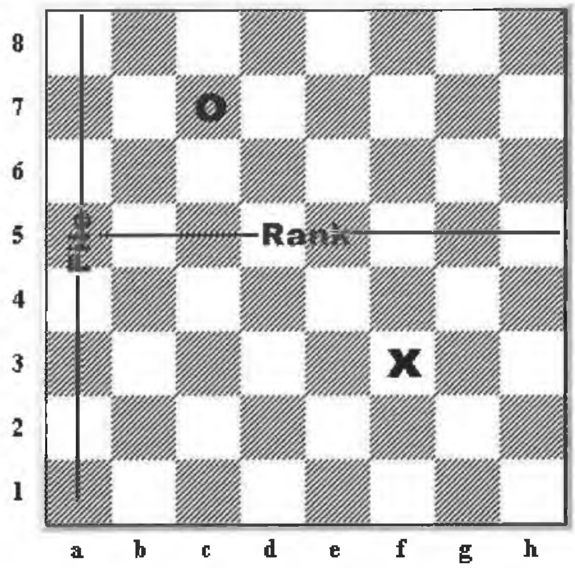
11.1 Unless announced otherwise in advance, a player who wins his game, or wins by forfeit, scores one point (1), a player who loses his game, or forfeits scores no points (0) and a player who draws his game scores a half point (1/2).

# Appendix B
## Chess Notation
*(Taken from http://www.chesscorner.com/)*

The moves of a chess game can be recorded in a variety of ways. You will probably see algebraic notation used more often but older chess books often use descriptive notation. It is a good idea to be conversant with them both. Chess positions can be recorded using Forsyth notation.

**Full Algebraic Notation**

The rows of squares on the chessboard are called ranks and the columns of squares are called files. The ranks are labelled from 1 to 8 and the files are labelled from a - h. We use these numbers and letters to describe where pieces are on the chessboard. In the diagram the blue cross is on the squared named f3 and the circle is on c7. Notice how the letter always comes first and the number follows it.




There are some symbols you should know when reading or writing chess notation.

| Symbol | Meaning | Symbol | Meaning |
|---|---|---|---|
| K | King | Q | Queen |
| R | Rook | B | Bishop |
| N | Knight | x | Captures |
| + | Check | ++ or # | Checkmate |
| O-O | Castles King's side | O-O-O | Castles Queen's side |

If you play in tournaments you will have to record the game so it is a good idea to practise as soon as you begin playing. You can also later go over your games to find out where you or your opponent made mistakes.

The moves are written in two numbered vertical columns like this:

| | |
|---|---|
| 1.f2-f4 | e7-e5 |
| 2.f4xe5 | d7-d6 |
| 3.e5xd6 | Bf8xd6 |
| 4.g2-g3 | Qd8-g5 |
| 5.Ng1-f3 | Qg5xg3+ |
| 6.h2xg3 | Bd6xg3# |

The first column is for the White moves and the second column is for the Black moves. First of all the symbol for the piece is written, then the square on which this piece was standing, then a hyphen (-), then the square to which this piece moves. If a pawn moves the symbol is omitted.

For example, 1. f2-f4 means on the first move the pawn on the f2 square moved to the f4 square. 5. Ng1-f3 means the Knight on the g1 square moved to the f3 square.

If you wish to refer to a Black move by itself you put three dots before the move. For example, 4. ... Qd8-g5 means on move 4 Black moved his Queen on d8 to g5.

x indicates a capture took place so: 5. ... Qg5xg3+ means the Black Queen on g5 captured a piece on g3 and the + means with this move the opponent's King was checked.

# means checkmate so: 6. ...Bd6xg3# means the Black Bishop on d6 moved to g3 and checkmated the White King.
This is what this game would look like on the chessboard:

1. f2-f4 e7-e5

2. f4xe5 d7-d6

2. e5xd6 Bf8xd6

3. g2-g3 Qd8-g5

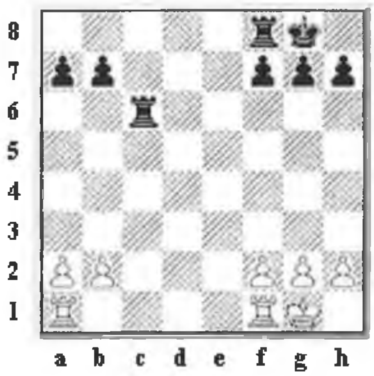
4. Ng1-f3 Qg5xg3+

5. h2xg3 Bd6xg3#

**Abbreviated Algebraic Notation**

In this type of notation the starting square of the chess piece is left out and only the destination square is written. If a pawn makes a capture then the file on which the pawn was standing is indicated.

In the diagram on the right, both the White Rooks can move to d1. To make it clear which one moves, the file on which the piece stands before it moves is indicated.



The diagram shows the Rook on f1 has moved to d1. This is written Rfd1.

Sometimes it may be possible that two pieces on the same file can move to the same square.

In the diagram on the right, both the Rooks can move to the d5 square. To show which Rook moves there we indicate the rank the Rook has moved from.

The diagram on the right shows the Rook on the seventh rank has moved to d5. We write this as R7d5.

This is how the game above would be written in short algebraic notation:

| | |
|---|---|
| 1. f4 e5 | The White pawn moves to f4 and the Black pawn to e5. |
| 2. fxe5 d6 | The White pawn on the f file takes the pawn on e5. The Black pawn moves to d6. |
| 3. exd6 Bxd6 | The White pawn on the e file takes the pawn on d6. The Black Bishop takes the pawn on d6. |
| 4. g3 Qg5 | The White pawn moves to g3. The Black Queen moves to g5. |
| 5. Nf3 Qxg3+ | The White Knight moves to f3. The Black Queen takes the pawn on g3 and checks the White King. |
| 6. hxg3 Bxg3# | The White pawn on the h file takes the Queen on g3. The Black Bishop takes the pawn on g3 and delivers checkmate. |

# Appendix C
Test Set Opening Sequences

| No. | Opening Sequence | Moves |
|---|---|---|
| 1 | Centre - 1 | 1. e4 e5 2. d4 exd4 3. Qxd4 Nc6 4. Qe3 Nf6 5. Nc3 Bb4 6. Bd2 O-O 7. O-O-O Re8 |
| 2 | Centre - 2 | 1. e4 e5 2. d4 exd4 3. Qxd4 Nc6 4. Qe3 g6 5. Bd2 Bg7 6. Nc3 Nf6 7. e5 Ng4 |
| 3 | French Defence - 1 | 1. e4 e6 2. d4 d5 3. Nc3 Nf6 4. Bg5 Be7 5. e5 Nfd7 6. Bxe7 Qxe7 7. Qd2 O-O |
| 4 | French Defence - 2 | 1. e4 e6 2. d4 d5 3. Nd2 Nc6 4. Ngf3 Nf6 5. e5 Nd7 6. Nb3 f6 7. Bb5 a6 |
| 5 | Kings Indian Defence - 1 | 1. d4 Nf6 2. c4 g6 3. Nc3 Bg7 4. e4 d6 5. Nf3 O-O 6. Be2 e5 7. O-O Nc6 |
| 6 | Kings Indian Defence - 2 | 1. d4 Nf6 2. c4 g6 3. Nc3 Bg7 4. Nf3 d6 5. Bf4 c6 6. e3 Qa5 7. Bd3 Nh5 |
| 7 | Nimzo Indian Defence - 1 | 1. d4 Nf6 2. c4 e6 3. Nc3 Bb4 4. Qc2 d5 5. a3 Bxc3+ 6. Qxc3 Nc6 7. Nf3 Ne4 |
| 8 | Nimzo Indian Defence - 2 | 1. d4 Nf6 2. c4 e6 3. Nc3 Bb4 4. e3 O-O 5. Ne2 d5 6. a3 Be7 7. cxd5 exd5 |
| 9 | Queens Gambit Accepted - 1 | 1. d4 d5 2. c4 dxc4 3. Nf3 Nf6 4. e3 e6 5. Bxc4 c5 6. O-O a6 7. Qe2 Nc6 |
| 10 | Queens Gambit Accepted - 2 | 1. d4 d5 2. c4 dxc4 3. e3 e5 4. Bxc4 exd4 5. exd4 Bb4+ 6. Nc3 Nf6 7. Nf3 O-O 8. O-O Bg4 |
| 11 | Queens Gambit Declined - 1 | 1. d4 d5 2. c4 e6 3. Nc3 Nf6 4. Bg5 Be7 5. e3 O-O 6. Nf3 Nbd7 7. Rc1 c6 |
| 12 | Queens Gambit Declined - 2 | 1. d4 d5 2. c4 e6 3. Nc3 c5 4. cxd5 exd5 5. dxc5 d4 6. Na4 b5 7. cxb6 axb6 |
| 13 | Queens Indian Defence - 1 | 1. d4 Nf6 2. c4 e6 3. Nf3 b6 4. g3 Bb7 5. Bg2 Be7 6. O-O O-O 7. Nc3 Ne4 |
| 14 | Queens Indian Defence - 2 | 1. d4 Nf6 2. Nf3 b6 3. g3 g6 4. Bg2 Bb7 5. c4 Bg7 6. O-O O-O 7. Nc3 Ne4 |
| 15 | Queens Pawn Games - 1 | 1. d4 d5 2. Nf3 Nf6 3. e3 c5 4. Nbd2 Nc6 5. c3 e6 6. Bd3 Bd6 7. O-O O-O |
| 16 | Queens Pawn Games - 2 | 1. d4 Nf6 2. Nf3 e6 3. Bg5 h6 4. Bh4 g5 5. Bg3 Ne4 6. Nfd2 Nxg3 7. hxg3 d5 |
| 17 | Reti Opening - 1 | 1. Nf3 d5 2. c4 d4 3. e3 Nc6 4. exd4 Nxd4 5. Nxd4 Qxd4 6. Nc3 e5 7. d3 Bc5 |
| 18 | Reti Opening - 2 | 1. Nf3 d5 2. g3 c5 3. Bg2 Nc6 4. O-O e6 5. d3 g6 6. Nc3 Bg7 7. a3 Nge7 |
| 19 | Ruy Lopez - 1 | 1. e4 e5 2. Nf3 Nc6 3. Bb5 Bc5 4. c3 f5 5. d4 fxe4 6. Bxc6 dxc6 7. Nxe5 Bd6 |
| 20 | Ruy Lopez - 2 | 1. e4 e5 2. Nf3 Nc6 3. Bb5 a6 4. Ba4 d6 5. c3 f5 6. d4 fxe4 7. Nxe5 dxe5 |
| 21 | Scotch - 1 | 1. e4 e5 2. Nf3 Nc6 3. d4 exd4 4. Nxd4 Bc5 5. Be3 Qf6 6. Nb5 Bxe3 7. fxe3 Qh4+ |
| 22 | Scotch - 2 | 1. e4 e5 2. Nf3 Nc6 3. d4 exd4 4. Nxd4 Nf6 5. Nc3 Bb4 6. Nxc6 bxc6 7. Bd3 d5 |
| 23 | Sicilian Defence - 1 | 1. e4 c5 2. Nf3 d6 3. d4 cxd4 4. Nxd4 Nf6 5. Nc3 a6 6. Bg5 e6 7. f4 Qb6 8. Qd2 Qxb2 9. Rb1 Qa3 |
| 24 | Sicilian Defence - 2 | 1. e4 c5 2. Nf3 Nc6 3. d4 cxd4 4. Nxd4 g6 5. Nc3 Bg7 6. Be3 Nf6 7. Bc4 d6 |

# Appendix D
## Test Results

### Test Setup 1
*MPSM 1*

| No. | Opening Sequence | Result |
|---|---|---|
| 1 | Centre - 1 | Draw |
| 2 | Centre - 2 | White |
| 3 | French Defence - 1 | White |
| 4 | French Defence - 2 | White |
| 5 | Kings Indian Defence - 1 | White |
| 6 | Kings Indian Defence - 2 | Draw |
| 7 | Nimzo Indian Defence - 1 | White |
| 8 | Nimzo Indian Defence - 2 | White |
| 9 | Queens Gambit Accepted - 1 | Draw |
| 10 | Queens Gambit Accepted - 2 | White |
| 11 | Queens Gambit Declined - 1 | Draw |
| 12 | Queens Gambit Declined - 2 | White |
| 13 | Queens Indian Defence - 1 | White |
| 14 | Queens Indian Defence - 2 | White |
| 15 | Queens Pawn Games - 1 | Draw |
| 16 | Queens Pawn Games - 2 | Draw |
| 17 | Reti Opening - 1 | Black |
| 18 | Reti Opening - 2 | White |
| 19 | Ruy Lopez - 1 | Draw |
| 20 | Ruy Lopez - 2 | White |
| 21 | Scotch - 1 | Black |
| 22 | Scotch - 2 | Draw |
| 23 | Sicilian Defence - 1 | Draw |
| 24 | Sicilian Defence - 2 | Draw |

| Result | Result Occurrence | Points per Result | Result Points |
|---|---|---|---|
| White Win | 12 | 0 | 0 |
| Black Win | 2 | 1 | 2 |
| Draw | 10 | 0.5 | 5 |
| | | **Total** | 7 |

# Test Setup 2

*MPSM 2*

| No. | Opening Sequence | Result |
|---|---|---|
| 1 | Centre – 1 | White |
| 2 | Centre – 2 | White |
| 3 | French Defence - 1 | White |
| 4 | French Defence - 2 | White |
| 5 | Kings Indian Defence - 1 | Draw |
| 6 | Kings Indian Defence - 2 | Draw |
| 7 | Nimzo Indian Defence - 1 | Draw |
| 8 | Nimzo Indian Defence - 2 | White |
| 9 | Queens Gambit Accepted - 1 | White |
| 10 | Queens Gambit Accepted - 2 | White |
| 11 | Queens Gambit Declined - 1 | Draw |
| 12 | Queens Gambit Declined - 2 | Draw |
| 13 | Queens Indian Defence - 1 | Draw |
| 14 | Queens Indian Defence - 2 | Draw |
| 15 | Queens Pawn Games - 1 | Draw |
| 16 | Queens Pawn Games - 2 | Draw |
| 17 | Reti Opening - 1 | Draw |
| 18 | Reti Opening - 2 | White |
| 19 | Ruy Lopez - 1 | White |
| 20 | Ruy Lopez - 2 | White |
| 21 | Scotch - 1 | Black |
| 22 | Scotch - 2 | White |
| 23 | Sicilian Defence - 1 | White |
| 24 | Sicilian Defence - 2 | Draw |

| Result | Result Occurrence | Points per Result | Result Points |
|---|---|---|---|
| White Win | 12 | 0 | 0 |
| Black Win | 1 | 1 | 1 |
| Draw | 11 | 0.5 | 5.5 |
| | | **Total** | **6.5** |

# Test Setup 3

*MPSM 1, END-GAME*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | W | 0 |
| 2 | Centre - 2 | END-GAME | W | W | 0 |
| 3 | French Defence - 1 | END-GAME | W | W | 0 |
| 4 | French Defence - 2 | END-GAME | W | W | 0 |
| 5 | Kings Indian Defence - 1 | END-GAME | W | W | 0 |
| 6 | Kings Indian Defence - 2 | END-GAME | W | W | 0 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | W | D | 0.25 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | W | W | 0 |
| 9 | Queens Gambit Accepted - 1 | Draw | W | W | 0 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | B | D | 0.75 |
| 11 | Queens Gambit Declined - 1 | Draw | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | W | W | 0 |
| 13 | Queens Indian Defence - 1 | END-GAME | W | W | 0 |
| 14 | Queens Indian Defence - 2 | END-GAME | W | D | 0.25 |
| 15 | Queens Pawn Games - 1 | END-GAME | B | B | 1 |
| 16 | Queens Pawn Games - 2 | END-GAME | W | D | 0.25 |
| 17 | Reti Opening - 1 | END-GAME | D | D | 0.5 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | Black | | | 1 |
| 20 | Ruy Lopez - 2 | END-GAME | W | W | 0 |
| 21 | Scotch - 1 | Draw | W | W | 0 |
| 22 | Scotch - 2 | END-GAME | W | W | 0 |
| 23 | Sicilian Defence - 1 | Draw | B | B | 1 |
| 24 | Sicilian Defence - 2 | White | | | 0 |
| | | | | **Total** | **6** |

# Test Setup 4

*MPSM 2, END-GAME*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | W | 0 |
| 2 | Centre - 2 | White | | | 0 |
| 3 | French Defence - 1 | END-GAME | W | W | 0 |
| 4 | French Defence - 2 | White | | | 0 |
| 5 | Kings Indian Defence - 1 | Draw | W | W | 0 |
| 6 | Kings Indian Defence - 2 | END-GAME | B | B | 1 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | W | W | 0 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | W | W | 0 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | W | W | 0 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | W | W | 0 |
| 11 | Queens Gambit Declined - 1 | END-GAME | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | B | B | 1 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | END-GAME | D | D | 0.5 |
| 15 | Queens Pawn Games - 1 | END-GAME | B | W | 0.5 |
| 16 | Queens Pawn Games - 2 | END-GAME | W | W | 0 |
| 17 | Reti Opening - 1 | END-GAME | B | B | 1 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | END-GAME | W | W | 0 |
| 20 | Ruy Lopez - 2 | END-GAME | W | W | 0 |
| 21 | Scotch - 1 | Black | | | 1 |
| 22 | Scotch - 2 | END-GAME | W | W | 0 |
| 23 | Sicilian Defence - 1 | END-GAME | D | D | 0.5 |
| 24 | Sicilian Defence - 2 | Draw | D | B | 0.75 |
| | | | | **Total** | **8.25** |

# Test Setup 5

*MPSM 3*

| No. | Opening Sequence | Result |
|---|---|---|
| 1 | Centre – 1 | White |
| 2 | Centre – 2 | White |
| 3 | French Defence - 1 | White |
| 4 | French Defence - 2 | White |
| 5 | Kings Indian Defence - 1 | White |
| 6 | Kings Indian Defence - 2 | Draw |
| 7 | Nimzo Indian Defence - 1 | Black |
| 8 | Nimzo Indian Defence - 2 | White |
| 9 | Queens Gambit Accepted - 1 | White |
| 10 | Queens Gambit Accepted - 2 | White |
| 11 | Queens Gambit Declined - 1 | Draw |
| 12 | Queens Gambit Declined - 2 | Draw |
| 13 | Queens Indian Defence - 1 | White |
| 14 | Queens Indian Defence - 2 | Draw |
| 15 | Queens Pawn Games - 1 | White |
| 16 | Queens Pawn Games - 2 | Draw |
| 17 | Reti Opening - 1 | Black |
| 18 | Reti Opening - 2 | White |
| 19 | Ruy Lopez - 1 | White |
| 20 | Ruy Lopez - 2 | White |
| 21 | Scotch - 1 | Black |
| 22 | Scotch - 2 | White |
| 23 | Sicilian Defence - 1 | White |
| 24 | Sicilian Defence - 2 | Draw |

| Result | Result Occurrence | Points per Result | Result Points |
|---|---|---|---|
| White Win | 15 | 0 | 0 |
| Black Win | 3 | 1 | 3 |
| Draw | 6 | 0.5 | 3 |
| | | **Total** | **6** |

## Test Setup 6

*MPSM 3, END-GAME*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | W | 0 |
| 2 | Centre - 2 | White | | | 0 |
| 3 | French Defence - 1 | END-GAME | W | W | 0 |
| 4 | French Defence - 2 | END-GAME | W | W | 0 |
| 5 | Kings Indian Defence - 1 | END-GAME | W | W | 0 |
| 6 | Kings Indian Defence - 2 | END-GAME | B | B | 1 |
| 7 | Nimzo Indian Defence - 1 | Black | | | 1 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | D | D | 0.5 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | | | |
| 10 | Queens Gambit Accepted - 2 | END-GAME | B | W | 0.5 |
| 11 | Queens Gambit Declined - 1 | END-GAME | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | D | W | 0.75 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | END-GAME | D | W | 0.25 |
| 15 | Queens Pawn Games - 1 | END-GAME | D | D | 0.5 |
| 16 | Queens Pawn Games - 2 | END-GAME | D | D | 0.5 |
| 17 | Reti Opening - 1 | END-GAME | B | B | 1 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | END-GAME | W | W | 0 |
| 20 | Ruy Lopez - 2 | END-GAME | W | W | 0 |
| 21 | Scotch - 1 | Black | | | 1 |
| 22 | Scotch - 2 | END-GAME | W | W | 0 |
| 23 | Sicilian Defence - 1 | END-GAME | W | W | 0 |
| 24 | Sicilian Defence - 2 | END-GAME | W | B | 0.5 |
| | | | | **Total** | **9.5** |

## Test Setup 7

*MPSM 4, END-GAME*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | W | 0 |
| 2 | Centre - 2 | White | | | 0 |
| 3 | French Defence - 1 | Draw | B | B | 1 |
| 4 | French Defence - 2 | END-GAME | D | D | 0.5 |
| 5 | Kings Indian Defence - 1 | Draw | W | W | 0 |
| 6 | Kings Indian Defence - 2 | Draw | D | D | 0.5 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | B | B | 1 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | B | D | 0.75 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | W | W | 0 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | W | W | 0 |
| 11 | Queens Gambit Declined - 1 | Black | | | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | B | B | 1 |
| 13 | Queens Indian Defence - 1 | Draw | D | W | 0.25 |
| 14 | Queens Indian Defence - 2 | Draw | | | |
| 15 | Queens Pawn Games - 1 | Draw | B | D | 0.75 |
| 16 | Queens Pawn Games - 2 | END-GAME | W | W | 0 |
| 17 | Reti Opening - 1 | Draw | W | W | 0 |
| 18 | Reti Opening - 2 | END-GAME | D | D | 0.5 |
| 19 | Ruy Lopez - 1 | END-GAME | B | B | 1 |
| 20 | Ruy Lopez - 2 | Draw | W | W | 0 |
| 21 | Scotch - 1 | END-GAME | B | B | 1 |
| 22 | Scotch - 2 | END-GAME | W | B | 0.5 |
| 23 | Sicilian Defence - 1 | END-GAME | B | B | 1 |
| 24 | Sicilian Defence - 2 | END-GAME | W | W | 0 |
| | | | | **Total** | **10.75** |

# Test Setup 8

*MPSM 1, END-GAME, Quiescence Search*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | B | B | 1 |
| 2 | Centre - 2 | END-GAME | W | W | 0 |
| 3 | French Defence - 1 | END-GAME | W | W | 0 |
| 4 | French Defence - 2 | END-GAME | W | W | 0 |
| 5 | Kings Indian Defence - 1 | END-GAME | D | D | 0.5 |
| 6 | Kings Indian Defence - 2 | END-GAME | B | B | 1 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | W | W | 0 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | W | W | 0 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | B | B | 1 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | D | D | 0.5 |
| 11 | Queens Gambit Declined - 1 | END-GAME | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | W | W | 0 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | END-GAME | W | W | 0 |
| 15 | Queens Pawn Games - 1 | END-GAME | W | W | 0 |
| 16 | Queens Pawn Games - 2 | END-GAME | W | D | 0.25 |
| 17 | Reti Opening - 1 | Draw | B | D | 0.75 |
| 18 | Reti Opening - 2 | END-GAME | B | D | 0.75 |
| 19 | Ruy Lopez - 1 | END-GAME | B | B | 1 |
| 20 | Ruy Lopez - 2 | END-GAME | D | D | 0.5 |
| 21 | Scotch - 1 | END-GAME | B | B | 1 |
| 22 | Scotch - 2 | END-GAME | W | W | 0 |
| 23 | Sicilian Defence - 1 | END-GAME | W | W | 0 |
| 24 | Sicilian Defence - 2 | END-GAME | D | D | 0.5 |
| | | | | **Total** | **10.75** |

# Test Setup 9

*MPSM 2, END-GAME, Quiescence Search*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | B | 0.5 |
| 2 | Centre - 2 | END-GAME | D | W | 0.25 |
| 3 | French Defence - 1 | Draw | W | W | 0 |
| 4 | French Defence - 2 | END-GAME | W | W | 0 |
| 5 | Kings Indian Defence - 1 | END-GAME | D | D | 0.5 |
| 6 | Kings Indian Defence - 2 | END-GAME | D | D | 0.5 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | W | D | 0.25 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | B | B | 1 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | W | W | 0 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | D | B | 0.75 |
| 11 | Queens Gambit Declined - 1 | END-GAME | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | B | B | 1 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | END-GAME | D | D | 0.5 |
| 15 | Queens Pawn Games - 1 | END-GAME | D | D | 0.5 |
| 16 | Queens Pawn Games - 2 | END-GAME | W | W | 0 |
| 17 | Reti Opening - 1 | END-GAME | W | W | 0 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | END-GAME | B | B | 1 |
| 20 | Ruy Lopez - 2 | END-GAME | B | D | 0.75 |
| 21 | Scotch - 1 | END-GAME | W | W | 0 |
| 22 | Scotch - 2 | END-GAME | D | D | 0.5 |
| 23 | Sicilian Defence - 1 | END-GAME | W | W | 0 |
| 24 | Sicilian Defence - 2 | END-GAME | B | B | 1 |
| | | | | **Total** | **11** |

## Test Setup 10

*MPSM 3, END-GAME, Quiescence Search*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | B | 0.5 |
| 2 | Centre - 2 | END-GAME | W | W | 0 |
| 3 | French Defence - 1 | END-GAME | W | W | 0 |
| 4 | French Defence - 2 | END-GAME | D | D | 0.5 |
| 5 | Kings Indian Defence - 1 | END-GAME | B | B | 1 |
| 6 | Kings Indian Defence - 2 | END-GAME | B | B | 1 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | B | B | 1 |
| 8 | Nimzo Indian Defence - 2 | Draw | W | B | 0.5 |
| 9 | Queens Gambit Accepted - 1 | Draw | B | B | 1 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | D | D | 0.5 |
| 11 | Queens Gambit Declined - 1 | Draw | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | D | D | 0.5 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | END-GAME | D | B | 0.75 |
| 15 | Queens Pawn Games - 1 | END-GAME | W | W | 0 |
| 16 | Queens Pawn Games - 2 | END-GAME | B | B | 1 |
| 17 | Reti Opening - 1 | END-GAME | W | W | 0 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | END-GAME | B | B | 1 |
| 20 | Ruy Lopez - 2 | END-GAME | B | B | 1 |
| 21 | Scotch - 1 | END-GAME | B | B | 1 |
| 22 | Scotch - 2 | END-GAME | D | D | 0.5 |
| 23 | Sicilian Defence - 1 | END-GAME | B | B | 1 |
| 24 | Sicilian Defence - 2 | END-GAME | D | D | 0.5 |
| | | | | Total | 15.25 |

## Test Setup 11

*MPSM 4, END-GAME, Quiescence Search*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | B | B | 1 |
| 2 | Centre - 2 | END-GAME | W | W | 0 |
| 3 | French Defence - 1 | END-GAME | D | D | 0.5 |
| 4 | French Defence - 2 | Black | | | 1 |
| 5 | Kings Indian Defence - 1 | END-GAME | B | B | 1 |
| 6 | Kings Indian Defence - 2 | END-GAME | B | B | 1 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | D | D | 0.5 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | W | W | 0 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | D | B | 0.75 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | D | B | 0.75 |
| 11 | Queens Gambit Declined - 1 | END-GAME | D | D | 0.5 |
| 12 | Queens Gambit Declined - 2 | Black | | | 1 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | END-GAME | W | W | 0 |
| 15 | Queens Pawn Games - 1 | Draw | B | B | 1 |
| 16 | Queens Pawn Games - 2 | END-GAME | B | B | 1 |
| 17 | Reti Opening - 1 | END-GAME | D | D | 0.5 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | END-GAME | B | B | 1 |
| 20 | Ruy Lopez - 2 | END-GAME | B | D | 0.75 |
| 21 | Scotch - 1 | END-GAME | W | W | 0 |
| 22 | Scotch - 2 | END-GAME | B | B | 1 |
| 23 | Sicilian Defence - 1 | Black | | | 1 |
| 24 | Sicilian Defence - 2 | END-GAME | W | B | 0.5 |
| | | | | Total | 15.75 |

## Test Setup 12

*NvN, END-GAME*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | END-GAME | W | W | 0 |
| 2 | Centre - 2 | END-GAME | W | W | 0 |
| 3 | French Defence - 1 | END-GAME | W | W | 0 |
| 4 | French Defence - 2 | Draw | W | W | 0 |
| 5 | Kings Indian Defence - 1 | Black | | | 1 |
| 6 | Kings Indian Defence - 2 | Draw | W | W | 0 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | W | W | 0 |
| 8 | Nimzo Indian Defence - 2 | END-GAME | B | B | 1 |
| 9 | Queens Gambit Accepted - 1 | Black | | | 1 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | W | W | 0 |
| 11 | Queens Gambit Declined - 1 | END-GAME | W | W | 0 |
| 12 | Queens Gambit Declined - 2 | Draw | B | B | 0 |
| 13 | Queens Indian Defence - 1 | END-GAME | B | B | 1 |
| 14 | Queens Indian Defence - 2 | Black | | | 1 |
| 15 | Queens Pawn Games - 1 | White | | | 0 |
| 16 | Queens Pawn Games - 2 | END-GAME | B | B | 1 |
| 17 | Reti Opening - 1 | White | | | 0 |
| 18 | Reti Opening - 2 | END-GAME | W | W | 0 |
| 19 | Ruy Lopez - 1 | END-GAME | B | B | 1 |
| 20 | Ruy Lopez - 2 | END-GAME | W | W | 0 |
| 21 | Scotch - 1 | END-GAME | W | W | 0 |
| 22 | Scotch - 2 | Draw | D | D | 0.5 |
| 23 | Sicilian Defence - 1 | Black | | | 1 |
| 24 | Sicilian Defence - 2 | END-GAME | W | W | 0 |
| | | | | Total | 8.5 |

## Test Setup 13

*NvN, END-GAME, Quiescence Search*

| No. | Opening Sequence | Result | White 1st | Black 1st | Points |
|---|---|---|---|---|---|
| 1 | Centre - 1 | Draw | D | D | 0.5 |
| 2 | Centre - 2 | END-GAME | B | D | 0.75 |
| 3 | French Defence - 1 | END-GAME | B | B | 1 |
| 4 | French Defence - 2 | END-GAME | D | D | 0.5 |
| 5 | Kings Indian Defence - 1 | END-GAME | W | W | 0 |
| 6 | Kings Indian Defence - 2 | END-GAME | B | B | 1 |
| 7 | Nimzo Indian Defence - 1 | END-GAME | W | B | 0.5 |
| 8 | Nimzo Indian Defence - 2 | Draw | B | B | 1 |
| 9 | Queens Gambit Accepted - 1 | END-GAME | B | B | 1 |
| 10 | Queens Gambit Accepted - 2 | END-GAME | D | B | 0.75 |
| 11 | Queens Gambit Declined - 1 | Draw | B | B | 1 |
| 12 | Queens Gambit Declined - 2 | END-GAME | B | B | 1 |
| 13 | Queens Indian Defence - 1 | Draw | W | W | 0 |
| 14 | Queens Indian Defence - 2 | END-GAME | D | D | 0.5 |
| 15 | Queens Pawn Games - 1 | END-GAME | D | D | 0.5 |
| 16 | Queens Pawn Games - 2 | END-GAME | B | W | 0.5 |
| 17 | Reti Opening - 1 | END-GAME | B | B | 1 |
| 18 | Reti Opening - 2 | END-GAME | B | B | 1 |
| 19 | Ruy Lopez - 1 | END-GAME | W | W | 0 |
| 20 | Ruy Lopez - 2 | Draw | W | W | 0 |
| 21 | Scotch - 1 | END-GAME | B | B | 1 |
| 22 | Scotch - 2 | END-GAME | B | B | 1 |
| 23 | Sicilian Defence - 1 | Draw | B | B | 1 |
| 24 | Sicilian Defence - 2 | END-GAME | D | D | 0.5 |
| | | | | Total | 16 |