Computer Board Games of Africa

(Algorithms, Strategies & Rules)

Prof. Johnson Ihyeh Agbinya
Department of Computer Science
University of the Western Cape, Private Bg X17, Bellville 7535, South Africa
jagbinya@uwc.ac.za

This book is dedicated to my deceased parents and uncles Agbinya Awurwu, Onwu Awurwu, Abi Edeh, Anyuwogbu elders and youths alive who introduced me to the first experiences on African board games. I can still hear them count as they play "hohe", "heeyeh", "hata", "hene", "kiriwoh". Without them, this book may have never been written. Keep counting and thinking.

Table of Contents

Part One

Chapter One

Terminology

"M kolibwobwo ka yeh ikwojeh"
"I went in search of wilting mushrooms and returned instead with a bunch of gold"! Igede (Nigerian) proverb

Introduction

My earliest introduction to African board games occurred when I was a little child in a Nigerian village Anyuwogbu where I was born and bred. My uncles and village elders, usually buddies and age group mates would gather in the village-square "ojiya". Ojiya is centrally located as the meeting point for the village. In it the village meets to celebrated and delibrate on issues of common interest. As kids we also played soccer with balls made of ropes from the forest which we wrap together to form small soccer balls. Soccer at this bare ground without grass with other village youths was always a social event. We play soccer in the dry season here after school, during holidays and evenings to keep ourselves busy and occupied. During the rainy season we shift to the soccer field which we created for ourselves beside the Anyuwogbu Methodist church. Village elders would normally sit by the bottom of the huge 'okpakoh' (baobad) tree playing "echi" a version of African board games. It was an exciting moment to watch by the sidelines the strategies adopted and the intense attention they paid to their opponents' moves. It was only recently that I started to realise the significance of some of the phraseologies they used. For example, while waiting to make the next move, you would hear an elder repeat "hohe", "heeyeh", "hata", "heneh", "kiriwoh", meaning, one, two, three, four, ten! I recently realised what they were doing. They were counting in base five! Is this not how we count in base five? You reach a count of four and then the next count becomes a 10! Much more than counting in base five, they were also estimating where the captured seeds in their hands would reach on the board when the sow the seeds. To ensure the opponent did not know how many seeds they possess in hand, they would normally count either two at a time or three at a time to confuse the opponent. Realising the significance of these strategies set me on a course to finding out for myself how much mathematical knowledge they were using in playing their games and whether there is anything to be learned by modern young Africans from these board games. No sooner have I started than I began to discover the huge interest on African board games by none Africans on the Internet. I think began gathering all the information available to me on the board games. Later the same year, I introduced our first project on African board games at the honours level at the University of the Western Cape, Cape Town, South Africa. Andre Henney conducted this research developing our first Java code for a 3x3 board game for a Nokia handset 6310.

I also came by anthropological books on the games and books that deal with the rules, but there was none to be found that really sets down the elements of mathematics involved in the games. Eventually when I conducted more research, I discovered several papers that have actually paid attention to the specific mathematics of the board games. This book contains a summary of such works and in many areas my own addition of new areas of mathematics that are used in playing the games. For example, we have introduced game defining matrices.

1.1 Fundamentals of African Board Games

African board games have distinct names, rules, strategies and styles that vary from region to region. Unfortunately for some reasons non-African writers have always lumped them together under the generic name of Mancala. Some authors also call them Manqala, Mankala or Mankaleh. This generic name is used for all versions of count and capture games. Mancala is actually a corruption of the word Mankaleh meaning the game of intelligence. It is hard to find the use of this name by African ethnic groups. Mankaleh is derived from Swahili a hybrid language that has a great deal of Arabic influence. Hence the name itself is at best a terminology adopted by Arabic traders to the continent. In this book, where necessary, we will use any of the Mankaleh derivative names. In general we prefer using the local names as used in the ethnic groups where each game is played or originates.

Since African board games are games of strategy, full information, logic and intelligence. Questions dealing with intelligence must be asked as we delve into their study. Logic and mathematical reasoning are also involved. As we strive to unravel the intricacies of the games, questions that boarder on intelligent reasoning need to be answered. Of the many questions that arise, the following are pertinent:

1. What mathematics can one learn from playing African board games?
2. Is an African board game one of strategy and logic only or does luck play a part?
3. Are there any strategies that seem to work all the time in playing African board games?
4. Does playing these board games improve retention of information and does it create other forms of intelligence?

In this book, answers to these questions will be attempted. First we will show some areas of mathematics that are directly used in the board games. You will find in the games elements of the following computing, mathematical algorithms and concepts:

1. Minimax algorithm and search
2. Never-Ending Moves
3. Evolutionary computation
4. Neural networking
5. Magic numbers
6. Heuristics
7. Decision theory
8. Game tree analysis
9. Game theory (winning positions)
10. Symmetry (rotational, reflectional or bilateral), periodicity, binary logic;
11. Probability concepts
12. Estimation theory
13. Sequence Analysis
14. Enumeration (combinatorics, Graph theory)
15. Alpha-Beta pruning
16. Limited peaceful search

None of these concepts is covered in detail. However, we have collated works on African board games using these concepts and algorithms as they become available to us. The list is not exhaustive and no pretensions are assumed either that we have gathered all of them.

We also show some of the strategies used to ensure successful outcomes and to immobilise opponents. These issues will be covered as they arise. The many versions of African board games are given in [11] below.

Figure 1: Distribution of Names of African Board Games

1.1.1 Who is Confused?

Similar to the confusion of names for the games, there is also an apparent 'confusion' of rules by publishers. In the minds and experiences of African owners of the games however, there are no confusions. The names are unique to regions and ethnic groups and each ethnic group has adhered to those names for centuries. Hence, we will not follow the international confusion of names and rules in this book. Rather, we will retain their local names and as much as possible their local rules as played by the owners of the games.

African traditional board games are as ancient as the continent itself and as varied as the cultural variation of its people. The rules of the games, the number of holes in each board and the number of seeds to sow per hole are varied. As such it has been hard to find a unified set of semantics or framework to describe them. This book attempts to organize these variations into a unified framework that makes distinction and playing of the games easier.

This book is also a race against time as many of the experienced and aged players die away, the need to document their held rules and expertise become extremely paramount. The spread of information on the Internet is also extremely scattered. Apart from a couple of sites that have compiled some of the rules and organised tournaments, the rest of the sites are extremely ill-organised. We also try in this book to bring some order to bear on the information on the games.

We will very often adopt a traditional approach and use the local names as much as possible with infusion of local terms that describe the games. Hence in this book there is preference for using African names and spellings.

The book is also a race to bring these games closer to African people of all ethnic groups. Most ethnic groups in Africa play a board game. Experience supports this notion. However, when it comes to recreational activities, we do not find many Africans playing board games. I have watched students playing games at the Union square at UWC in Cape Town, but rarely find them playing an African board game.

We are not going to be apologetic in any sense as authors of African games science and mathematics have been driven to be. Rather facts are stated without apology. Europeans and others from various parts of the world have never been apologetic in describing the level and state of science and mathematical knowledge that have origin in those continents. Hence we cannot and will not be apologetic. The millions of references in books and Internet on origins of contemporary mathematics and science points to contributions from all races, so that no race or continent can claim monopoly in laying foundations. Hence we write on the mathematics of African board games to the point.

We have taken a technical approach with a focus for identifying mathematical concepts and computing algorithms. Hence, anthropological slants are avoided. We do not even go into the history of these games because doing so will divert us from our focus greatly.

We do not claim ownership of all the information included in this book. However in significant portions of the book, we have originated the content from scratch, particularly in some of the mathematical models provided. Where the models fit and agree with the games, we wear the hat and develop it. There are two proofs on "never-ending moves" that are taken from the Internet directly from the works of other researchers. They are reproduced with very minor formatting that facilitates reading and understanding. We are particularly interested in materials that include mathematical formulations of the games as well as software implementations.

Africans in Diaspora, in the West Indies, South America and the Middle East have carried along their cultural heritage to where they are and play these games under various names and rules. We have included a few of such rules in this book and more of what actually is found in the continent as our focus is very much on the continent in this edition.

Traditional African board games have always been played between two human beings. The introduction of African games in software adds a new dimension. The first software on any African board games was written in Uganda by Professor J V Mayega, Department of Mathematics of Makere University Kampala in 1976. Other early works in African software games were based on Mankala [1, 2] and come mostly from Europe and the USA. They are mostly unavailable within the continent and as such not within the reach of traditional average African players.

To broaden the population of players and to create interest in these games beyond the village square settings, one must expose them to a broader audience in writing. This is one of the objectives of this book. It is also necessary to analyze them for reasoning and enhancement of interest and game potential. This goal is also pursued in this book as we document and collect both physical and software games into one repository for a larger playing audience.

1.1.2 Mathematical Strengths

African board games provide among others the following key strengths:

1. Strategic thinking
2. Employment of mathematical skills
3. Employment of logical skills
4. Ability to plan and forward thinking in playing
5. Ability for abstract thinking

African board games are games of intellect and thinking rather than chance. They require strategies and build character and mathematical thinking. In the words of one author, “pit and pebble games are probably the most arithmetical of all games”. African board games “can be introduced purely as a game of chance to very young children and even at this level, it has subtle educational value in encouraging the child to count. He or she progressively also learns the concept of one-to-one correspondence as he drops each one seed into each of a sequence of consecutive holes. Soon he learns simple sums in order to evaluate options and keep score”. We are doing more than this in this book.

For young as well as aged players as he advances in the discovery of the game the “player will begin to see the strategic importance of planning and the discipline involved in the actual implementation of long-term strategies appreciating the importance of foresight, correct timing and an awareness of the principle of cause and effect”[12].

1.2 Short List of Games from Africa

There is no difficulty in finding the many names of African board games. The Internet is full of them. There is however not too many concerted efforts to organise and relate them to their origins and their contributions to the mathematical thinking of Africans. This may be deliberate or an oversight. The following table is a summary of some of the board games and their countries of origin.

Game	Rows	Holes/Row	Seeds	Seeds /hole	Where Found	Descriptions
Ayo (Ayoayo)					Nigeria (Yoruba)	
Ayo J'duo					Nigeria (Yoruba)	
Aju					Togo	
Adji-Boto (Oware)					Ghana (Ewes)	
Awale	2	6	48		Senegal, Kenya (Masai), Ivory Coast	Board game
Awèlé	2	6	48		Ivory Coast, Ghana (Ga)	
Awarri	2	6	48			2 players
Oware	2	6	48		Ghana (Akan)	2 players
Oware (Abapa)	2	6	48		Ghana	Adult version
Oware (Nam-nam)	2	6	48			Children version
Oware (Tampoduo)	2	6	48		Ghana	Children version
Owari	2	6	48		Ghana, Burkina Fasso	
Ouri	2	6	48		Senegal, Cape Verde	
Warri	2	6	48		Barbados, Antigua	2 players
Kalah	2	6	72			
Gebeta	3	12		3	Ethiopia (Amharic), Eritrea	
Bao Zanzibar	4	8	64		Zanzibar	2 players
Bawo	4	8	64	4	Malawi	
Omweso	4	8	64	4	Uganda	2 players
Tchadji	4	8	64	2	Mozambique	
Bantumi					South Africa	
Jerin-jerin	2	6	48		Nigeria (Yoruba)	
Layli Goobaly	2	6			Somalia	
Mankala Mankaleh Mancala	2	6	36	3	Egypt (Arabic)	Board game; Before 1400 B.C
Moruba	4	36			South Africa (Limpopo)	Board game
Sideko					South Africa	
Soro					Tanzania	Board game
Kpo					Sierra Leone, Liberia	Board game
Echi	2				Nigeria (Benue, Igede)	Board game
Adi	2	6	48		Nigeria	
Ba-Aa					Ghana	
Mbau					Angola	
Nsa-Isong					Nigeria (Efik)	Board game
Pogu					Liberia (Bassa)	
Tsoro					Zimbabwe	
N'tchuva (Ncuva)					Mozambique, south africa	
Mpale					Mozambique	
Yote						

Achi					Ghana	
Shisima					Kenya	

Table 1: Names of Board Games of African origin

Some inferences can be drawn from this table. Board games found in the West Africa are mostly two rows, six column versions. In Central, East and Southern Africa, a lot of them are 4 rows and 8 column versions. Other extensions in the Southern African region include Moruba with 36 columns and 4 rows. Moruba probably has the most elaborate and complex board with 36 holes per row. Gemeta (Eritrea) in North Africa uses three rows. This too is distinctly different from the rest of African board games. The geographical distribution of the board games is shown on this Figure [2].

Table 2 provides a distribution of the names. It shows many of the names to be very close or almost the same, suggesting adaptation of names by various ethnic groups.

A	ABALALA'E, ABANGA, ABANGAH, ABOUGA, ACHARA, ADI, ADITA TA, ADITO, ADJI, ADJIBOTO ADJIKA, ADJI PRE, ADJITO, AGHI, AGI, AJI, AJWA, AKONG, ALÉ, ANDOT, ANNANA, ANYWOLI AWALE, AWALÉ, AWARE, AWARI, AWELE, AWÉLÉ AYO, AYO AYO, AZIGO
B	BA-AWA, BANGA, BANTUMI, BAO, BAO KISWAHILI, BAO SOLO, BARE, BARUMA, BAU, BAWO, BECHI, BOKE, BOSH, BOUBEROUKOU, BOURI
C	CHANKA, CHISOLO, CHONGKAK, CHORO, CHOUBA, CHUBA, CHUNCA, CISOLO, CONGKAK COO, CORO, CORO BAWO
D	DABUDA, DAKON, DAKOUN, DARA, DARRA, DEKA, DJONGHOK, DJONGLAK, DWONG
E	ECHI, ÉRHÉRHÉ, ENDODOI, ENKESHUI, ESON XORGOL, ESSON, ÉU LEU
F	FANGAYA, FUVA
G	GABATA, GABATTA, GALATJANG, GAMACHA, GBÉGÉLÉ, GEBTA, GELO, GEPETA, GESUWA GILBERTA, GIUTHI
H	HALUSA, HUS
I	IGISORO, IGOSOU, IKIOKOTO, IMBELECE, IMBWE, IMPERE, ISAFU, ISE ONZIN EGBE, ISOFU, ISOLO
J	J'ERIN, JODU, J'ODU, JUKURU
K	KACHIG, KA IA, KALAH, KALAHA, KALAK, KALE, KALIMANTA, KASONKO, KATRA, KBOO KENJI GUKI, KIARABU, KISOLO, KISWAHILIBAO, KIUTHI, KPO, KROUR, KUBUGUZA
L	LA'B HAKIM, LA'B MADJUNNI, LA'B ROSEYA, LAHEMAY WALIDA, LAMI, LAMLAMETA, LAMOSH LAM WALADACH, LANGA HOLO, LAYO, LEKA, LELA, LEYLA GOBALE, LIEN, LIZOLO, L'OB AKILA LONGBEU A CHA, LONTU HOLO, LUZOLO
M	MANCALA, MANDIARÉ, MANGA, MANGALA, MANGOLA, MANKALA, MANQALA, MANQUALA, MARABOUT MARANY, MARUBA, MAZAGEB, MBANGBI, MBAU, MBELETE, MBERE, MBO, MBOTHE, MEFUHVA MEFUVHA, MEUSUEB, MEWELAD, MOFUBA, MORO GBEGELE, MOTIQ, MSUWA, MULABALABA MUNGALA, MUTITEBA, MWAMBALULA, MWEISO, MWESO
N	NAKABILE, NAMBAYI, NARANJ, NCHOLOKOTO, NCHOMVWA, NCHUBA, NCHUWA, NDOTO, NGAR NJOMBWA, NOCHOLOKOTO, NSOLO, NSUMBI, NTCHUWA, NUMNUM
O	OKO, OLINDA, OKWE, OMWEESO, OMWESO, OTEP, OTJITOTO, OT JUN, OTRA, OT TJIN, OTU OURÉ, OURI, OURIN, OURRE, OURRI, OWARE, OWELA
P	PALANKULI, PALLAMKURIE, PALLAM KUZHI, PALLANGULI, PALLANKULI, PANDI,

PAPADAKON, PAPANDATA
PENSUR, PEREAUNI, PÉRÉSOUNI, POO

Q QALUTA, QASUTA, QELAT

R RYAKATI

S SADDEKA, SADEKA, SADIQA, SCHACH, SERATA, SIG, SOLO, SOMBI, SONGO, SORO, SPRETA
SULUS NISHTAW, SUNCA, SUNGKA

T TAGEGA, TAMTAM APACHI, TAP, TAPATA, TCHANKA, TCHOKAJON, TCHONKKAK, TCHOUKAITLON TCHUKARUMA
TEGRE, TJONGLAK, TOGUZ XORGOL, TOI, TONKA, TOPUZ XORGOL, TSCHUBA, TSH ELA, TSHUBA, TSHI SOLO, TSORO

U UBAO, UGWASI, UM EL BAGARA, UM EL BANAT, UM EL TUWEISAT, URDY, URÉ

V VAI LUNG THLAN

W WALÉ, WALLE, WALU, WALYA, WARE, WARI, WARRI, WAWEE, WAWI, WEG, WORI, WORIBO
WORO, WOURI, WULI, WURI

X XORGOL

Y YADA, YIT NURI, YOVODJI

Z

Table 2: Distribution of names of African board games

(Source: http://www.gamesacrosstheboard.com/welcome/main/content/rules/mancala/)

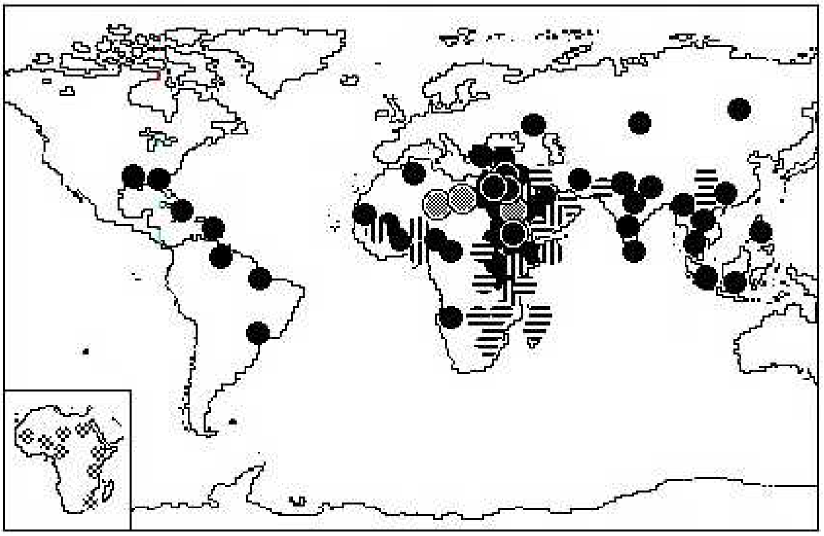


Figure 2: Geographical distribution of Board Games [10]

inset: distribution of the *dara* game

Neolithic mankala	**3-row mankala**
2-row mankala	**4-row mankala**

Source [10]

1.3 Terminology

Games are best enjoyed through using special terms. Terms excite, depict acts of play, success and loss playing games. This chapter attempts to capture a unified terminology for African board games. The words and phrases are assembled from various parts of the continent, articles and books [1-8]. Therefore you will find words from many African languages. In this book they are applied as a unified semantics or language that can be used to play all the board games. Therefore, you will find words from various African countries contributing to the overall terminology. Many of the terms listed are popularly applied to the Bao game in Zanzibar and derived from Swahili. Since board games are popular in the continent, there will be similar words in many other languages. Our aim is not preference, rather availability of relevant terms.

1.3.1 Description of Board Games

Bao is the board and by extension the game played on it

Kete (empiki) are the seeds used in playing any of the board games

Kichwa are the holes at the ends of each row. In a board of two rows, there are 4 kichwas. Similarly in a board of 4 rows there are 8 kichwas. The number of kichwa (k) in a board of r rows is given by the expression:

$$k = 2r;$$

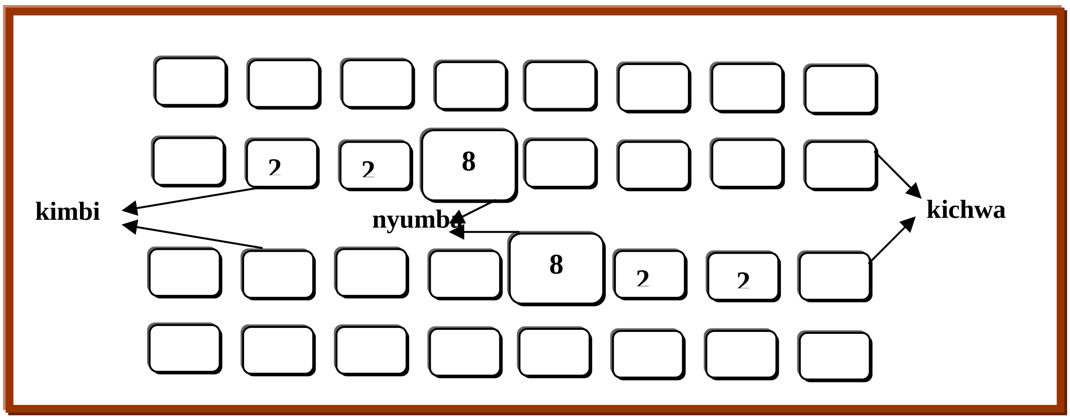


Figure 3: A Typical Board Game with Descriptions

Kimbi is the second holes from left and right of a board typically in the second row of most 8-hole rows. They are next to the Kichwa.

Kuu is also used to refer to the main hole with 8 pieces and special powers in the opening stage, until first moved. This is the Malawian equivalent name for Nyumba

Nemo refers to the process when a single seed is added in the opening stage

Nyumba - The nyumba (in Swahili means 'house') is the hole marked with a rectangle in a Bao game. This is always the fifth hole from the left on the front row. We will also use the word to mean the fifth hole in board games if a fifth hole exists. The nyumba will also be used to refer to the hole where captured seeds may be dropped as repository or barn. In the Bao game, the nyumba ceases to be a nyumba as soon as the seeds it contains are sown. After that it is an ordinary hole just as all other holes.

Consider a board game of 4 rows (2 rows per player) and represent each hole in the rows visually with the digit 0. The nyumba is represented with digit 1 (Figure 4) and other holes with 0. These are only visual representations and do not indicate how many seeds (kete) each hole contains.

0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0

Figure 4: Visual Representation of the Nyumba

1.3.2 Moves

All the ethnic groups have words that describe moves or playing turns.

Namua is the first stage of some board games when each player brings one seed into play each turn.

Mtaji is the stage after namua. A situation where if there is no winner during the first stage (namua), players keep on playing with the seeds on the board until there is a winner. This is called the Mtaji stage. It is also an attacking move that results into capturing opponent's seeds.

Mitaji – plural of mtaji

Kroo - A kroo is when a player has enough seeds to go all the way around the board

Okukonezebwa ("oh – koo- koo – neeza") – In the game of Omweso, this is moving three (3) seeds during the opening phase before first capture, 2 seeds into next hole and final seed into the next but one.

Takata is a move **not** resulting in capture

Piga Tanji – "To piga tanji is to attack (usually) two holes, both with many seeds, in one move [8]. This often forces an opponent to abandon one of them. If one of the holes is a kimbi hole, players usually abandon the kimbi hole to avoid kitakimbi."

"Kitakimbi is a trap that forces a player to defend his kimbi hole". Since it is difficult to defend that hole, the player abandons his other holes [8].

Relay sowing – this is the procedure in some board games where a player is allowed to continue playing as long as his last seed falls in a hole with seeds in it already. The player stops sowing and gives turn to the opponent once the last seed falls in an empty hole.

1.3.3 Capturing of Seeds

Hamna - In a board game of 2 or more rows per player, the clearance of the front row is called a hamna. In Bao, this ends the game and the player whose front row is empty is the loser.

Takasa – no capturing of a seed from your opponent (also used to refer to moves where you cannot start with a captured seed).

Utitiri - In some board games it is often wise to launch a singleton attack when many seeds from the front row have been captured. This strategy is called utitiri 'chicken lice'.

Emitwe-Ebiri – “cutting off at two heads”. This is capturing of both extreme pairs of holes in one move. The player is hemmed in.

Akawumbi – ('The Billion') This applies to Omweso. The opponent's seeds are initially in every hole in the board (although this situation is exceptionally rare, it could happen in a real game) and they are all captured in one move with the last holes captured being of a 'head'.

1.3.4 Winning

Akakyala (“akka-challa”) – In Omweso, this is capturing of seeds from the loser in two separate moves before the loser has made a first capture of the game.

Mkononi – winning a game during the namua, stage. You win mkononi ('in hand') because there are still seeds left in hand to bring into play.

Apart from variations in the terms used for describing board games of African origin, there are many contributing parameters to the variation of African board games. First, the number of rows, holes per row and the number seeds sowed per hole at start contribute to the variations. Second the direction of sowing is another contributor. Rules vary. These variations are explored in the subsequent chapters.

1.4 References

[1] Voogt, Alexander J. de, 1995 Limits of the Mind, towards a characterisation of Bao masterhip, Research School CNWS, Leiden, ISBN 90-7378249-X, Thesis Rijksuniversiteit Leiden
[2] Béart, C., 1955 Jeux en jouets de l'Ouest Africain, Dakar
[3] Murray, H.J.R., 1952 A History of Boardgames other than Chess, London
[4] Russ, L., 1984 Mankala Games, Michigan
[5] Townshend, P., 1979 Anthropological Perspectives on Bao (Mnkala) Games, University of Nairobi , Institue of African Studies Paper 114, Nairobi
[6] Voogt, A. J. de, 1997 Mankala Board Games, British Museum Press, ISBN 0-7141-2536-9, London
[7] Rob Nierse, “Introduction to Bao”, Waterlelieweg 56, 2215 GP voorhout, Holland
[8] Rob Nierse, “Tactics of Bao”, Waterlelieweg 56, 2215 GP voorhout, Holland
[9] van Binsbergen, W.M.J., 1997, Board-games and divination in global cultural history: a theoretical, comparative and historical perspective on mankala and geomancy in Africa and Asia; **http://www.shikanda.net/index.htm**
[10] van Binsbergen, W.M.J., 1997, “Rethinking Africa’s contribution to global cultural history II”
[11] The Oware Society, http://www.oware.org/rules.asp
[12] W Lee Farum-Badley, "Barbados Warri: the Bajan way"; http://barbadosphotogallery.com/warri/warrigame.htm].

Chapter 2

Algorithmic Representation

Introduction

Mathematical representation of games helps to transform them into algorithms that can be coded in software and aids their development. The algorithms also enhance mathematical thinking. The process of playing African board games can be represented by three mathematical operations, addition (sowing), subtraction (capturing seeds) and shifts (new state of the board). The process also involves matrices.

Matrices are used to represent the initial state of each game. Latter in this chapter we present the state-space representation. Consider the basic mini mankala game of two rows and 3 columns as coded by Henney and Agbinya [1].

A simple 2x3 matrix gives the initial state of this game.

$$\begin{Bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \end{Bmatrix}$$

Each of the two players owns a row of three cells with 2 seeds in each cell to use in playing. The total number of seeds is therefore 12. A game of 12 seeds is of low complexity. Sowing of seeds is counter-clockwise. A player can sow seeds into the opponent's cells. No cell or row can accumulate more than 7 seeds. In other words, the capacity of each cell or row is limited. This leads to winning because as soon as a player has two or less seeds in his row, he is the winner. Therefore the opponent must have accumulated about 10 seeds in his or her row.

2.1 A Move

Represent a move a discrete expression in which the number of holes reached by sowing new seeds is given by n and the x(k) denote the number of seeds in hole k at the beginning of iteration 1 and y(k) the number of seeds in hole k at the beginning of iteration 2. Therefore:

$$y(k) = x(k) + n$$

The content of a hole will increase by 1 due to sowing if the integral value n is large enough to reach the hole after depositing seeds in its predecessor holes.

2.2 Even Characteristics of African Board Games

The use of even number in African board games is pronounced. The following characteristics of all the board games involve even numbers:

- Rows of holes in the board games, usually two or four rows per board (except Gebeta)
- The number of holes per hole, usually, 2, 4, 6 or 8
- The number of seeds sown per hole at start of each game, usually 2 or 4 seeds per hole
- The number of players at a time, usually 2

2.3 Game Complexity

The complexity of a board game provides an indication of the degree of difficulty involved to play the game and to convert its rules into software format. Four views on the complexity of games can be expressed as:

- State – space complexity
- Game – tree complexity
- Mutational complexity and
- Computational complexity of the algorithms used in its software production

2.3.1 State-Space Complexity

The state-space of a board game is the various pattern of distribution of seeds on the board, from initiation to the end of the game. The current state space can be used as an input to an evaluation function that is used to evolve a game. The output of the evaluation function can also be used in a mini-max search.

The state-space and game-tree complexities of several African board games were addressed by Wernham [3] and this section is a summary of that work. The state-space complexity of a board game is defined here as the various ways (permutations) available for setting up the game at start, the number of moves to establish winning result, the variation in the number of illegal moves and complex moves. The expression defining state-space complexity combines the number of holes used in the game, the number of seeds involved and the number of illegal or improbable positions. The state-space complexity C(h, s, k) of African board games described in this book is:

$$C(h, s, k) = \frac{(h - 1 + s)!}{(h - 1)!\, s} - k$$

Where:

h = number of holes
s = number of seeds
k = number of illegal/improbable positions

For example, a board game with no illegal moves (k=0), 12 holes and 48 seeds has state-space complexity of 7.2381×10^{70}. If there are improbable positions in the game, the state-space complexity is reduced by that number. An improbable position in Oware would include 47 seeds in one hole, and the remaining seed in another.

The state-space complexities of several African board games are compared in Table 1. The initial set up complexity is unity for games like Oware (Awari) and Bao with unique seed contents in each hole.

Game	Initial set	As play starts	Midgame	Endgame
Awari	1	Slowly rising	2.8×10^{11}	Falling
Bao	1	Very slowly Rising	10^{25}	-
Omweso	5.6×10^{23}	Rising quickly	10^{25}	-

Table 3: State-space complexities of 'world' games – Adapted from [3]

In Omweso, each player has 32 seeds to set up, giving 7.5×10^{11} possible positions for the first player to set up, which can be countered by the other player in 7.5×10^{11} ways giving 5.6×10^{23} combinations [3]. As noted in [3], “in tournament play there are no illegal set-up combinations, and very few improbable ones.

The ‘k’ factor for mankala games is lower than in positional games since many pieces may share the same ‘hole’ in mankala. … State-space complexity in Omweso rises rapidly after the first captures, and remains high throughout as the seeds become redistributed in large numbers quasi-stochastically from one player to the other”.

There are implicit assumptions leading to the values in Table 1, therefore these values are approximations. Different games have different numbers of seeds in play at different stages. For example, in Bao re-entrancy is permitted and seeds are introduced throughout the initial phase of the game. Captured seeds are re-entered onto the board.

Games with high theoretical state-space complexity may be less ‘intricate’ for humans as the outcomes are beyond mental calculation and require ‘brute force’ calculations of no finesse [4]. This implies that the interesting games are games that are on the edge of human capacity for calculation and tactics and especially those that show high degrees of chaotic behaviour. This observation was initially made in [4].

2.3.2 Game Tree Complexity

Each board game forms a tree that evolves as the game is being played. This concept should be understood under the general semantics of data networks. A tree is formed when a series of operations lead to others and those steps can be represented pictorially as logical growth of the tree from one node to the next.

Game tree complexity provides an estimate of the number of branches in the tree at set-up used to represent the game based on a number of players, the branches per move and the length of the play.

Game-tree complexity can be calculated as:

$$G_t = i_1 \times i_2 \times b^p$$

Where,

i = branches in set-up of game for players 1 and 2

b = branches per move
p = plays in game length (average game length)

The search space is given by the number of branches raised to the power of the average length of the game.

The game tree complexity of Oware, Bao and Omweso are given in Table 2 [3]. The complexity values are affected by the state-space at initial set up. For example, Omweso has a large variation in the state-space at start up and leads to a large value of endgame permutations.

Game	Initial set	By endgame (no forced moves)
Awari	1	2.8×10^{32}
Bao	1	2×10^{34}
Omweso	5.6×10^{23}	$(5.6 \times 10^{23})(5 \times 10^{50}) = 2.8 \times 10^{74}$

Table 4: Game-tree complexities of African board games [12]

Another expression that can be used to estimate the game-tree complexity is given in [14] as the product:

$$G_t = \prod_{k=1}^{d} w(k)$$

where, w is the branching factor, and d the average length of the game. In this expression, every step of the game is modeled as starting from a node with w branching factors or w different options available to the player. With d=57 and w= 4 the game-tree complexity for Bao is given by the following:

Game	depth	Complexity (G_t)
random-play		2.8×10^{34}
random evaluation	4	3.1×10^{37}
random evaluation	10	1.5×10^{37}
fixed evaluation	4	5.0×10^{28}
fixed evaluation	10	5.7×10^{28}

Table 5: Game-Tree Complexity of Bao [5]

In Oware, feeding of the opponent's holes when they are all empty is permitted. This may be considered as a forced move. Omweso has nothing of that nature. However, in Bao forced moves are very common [3], perhaps 1/10 in master-level plays and more common amongst less experienced players who do not know how to avoid or take advantage of these situations.

In Bao a player has choices of where to re-enter captured seeds (left or right) during the initial stage of the game. In Omweso however, the player has choice of where to begin sowing to make a capture but not where to re-enter the seeds. These considerations must be made in estimating the game-tree complexity.

2.3.3 Mutational Complexity

Like in genetic mutations, the state spaces of games mutate. This concept was first pinpointed by De Voogt [4]. The mutational complexity of a game is the number of changes on the board due to a single move. Moves change the nature of the board by changing the state-space, the positions and the number of seeds in the holes due to planted new seeds.

In Bao capturing re-entry rules genetically change the nature of the front row as new seeds are planted there. Mutational complexity is less pronounced in Oware since there is no relay sowing. Therefore the impact of a move is limited to a few holes.

2.4 References

[1] Henney, Andre; Agbinya, Johnson; Ron Eglas, "African Board Games", Honours Project Report, Department of Computer Science, University of the Western Cape, 2003.

[2] Zaslavsky, Claudia. *Africa Counts*. Boston: Prindle, Weber & Schmidt inc. 1973

[3] Brian Wernham, “Omweso: The Royal Mankala Game of Uganda – A General Overview of Current Research”, International Omweso Society, http://www.omweso.org, 2002, pp. 1-26.

[4] De Voogt, “Limits of the mind”, Leiden, CNWS, 1995.

[5] Jeroen Donkers and Jos Uiterwijk, Programming Bao, IKAT, Dept. of Computer Science, Universiteit Maastricht, Maastricht, The Netherlands, 2003

Chapter 3

Mathematics of African Board Games

Introduction

Mathematics in African cultural games is varied, intricate and widespread. This section limits the discussions to the mathematics adopted in African board games. Several of them are discussed first explaining the methods and then how they are used in the games.

All African board games involve shift, add and subtract operations. Usually many of the moves can be represented with a function in which the numerical values increase by one at a time except one value that is decreased to zero entirely.

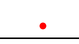
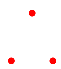
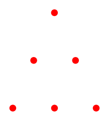
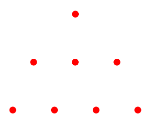
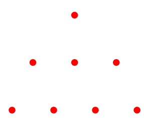
African board games are games of perfect information, otherwise known as combinatorial games. They are mostly games of two-players, with no hidden information, no chance moves, restricted outcome (win, lose and draw) and with each player moving alternately. This chapter covers some innate mathematics of the board games hitherto not widely recognized. In subsequent chapters, other areas of mathematics used in the games are covered.

3.1 Nonlinear Additive Series (Triangular Numbers in 'tarumbeta')

It can be shown that the game of tarumbeta in East Africa utilized the triangular numbers [2] or additive series (1, 3, 6, 10, 15 …). This series can be represented by the expression:

$$S(n+1) = S(n) + n; \quad S(0) = 0$$

The previous number plus the iteration count next number.

Index, n	**S(n)**	**S(n+1)**	**Triangle**
1	0	1	
2	1	3	
3	3	6	
4	6	10	
5	10	15	

			• • • • •
• • •	• • •	• • •	

Table 6: Triangular Magic Numbers

The numbers form an isosceles triangle. The next triangle is created by adding another layer of seeds to any side of the previous triangle and this number of seeds is the iteration count. The algorithm can be realized using the following diagram

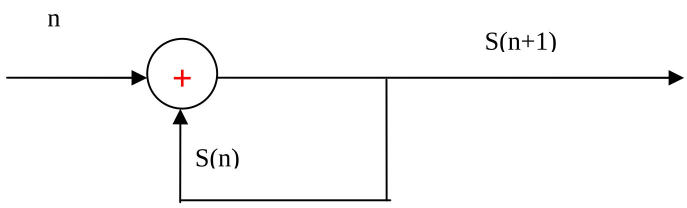


Figure 5: Generation of Triangular Numbers

3.2 Discrete Self-Organization in Oware

The board game oware is known by many names as echi, ayo, bao, giuthi, lela, mankala, omweso, owari, tei and songo. The game of Owari uses discrete self-organizing maps. The owari game with its two rows of 6 holes each is shown in Figure 5. At the beginning of each game, there are 4 seeds per hole.

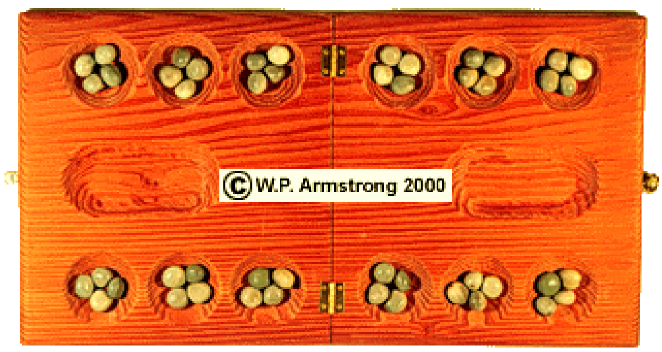


Figure 6: Oware (after W.P Amstrong)

Playing this game often result to recursive moves or self-propagating patterns, the so-called "marching group" moves by Ghanian players of the game. Provided the number of counts in a series of neighbouring holes decrease by unity (eg. 4, 3, 2, 1) the entire series can be replicated by a single right shift. This occurs by taking all the seeds in the hole with the largest number of seeds and sowing them into the next four holes. This pattern unless interrupted in some manner by an opposing player enables a player to avoid capture of seeds by simply moving the pattern one hole at a time. The pattern propagates round the board as long as the game remains. This situation is shown below in Figure 6, where scooping and sowing the four seeds will not change the pattern but merely propagates it one hole to the right.

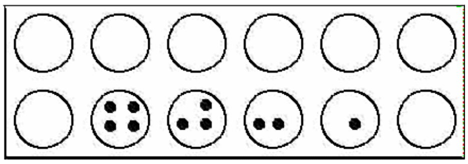


Figure 7: A Propagating Pattern In Oware

From Figure 5, it can be shown that if the contents of neighbouring holes are decimal digits arranged in reverse order of magnitude, a propagating pattern will result.

3.3 Cellular Automata

Oware was described in [1] as a one-dimensional cellular automation. How the self-propagating pattern described above was illustrated. For example, starting from 3 4 2 1 it can be shown that 13 iterations are required. The pattern is generated following the iteration:

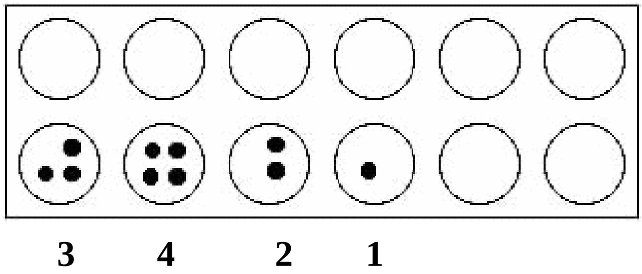


Figure 8: Generating a Self-Propagating Pattern

$$3\,4\,2\,1 \Rightarrow 5\,3\,2 \Rightarrow 4\,3\,1\,1\,1 \Rightarrow 4\,2\,2\,2 \Rightarrow 3\,3\,3\,1 \Rightarrow 4\,4\,2 \Rightarrow 5\,3\,1\,1 \Rightarrow 4\,2\,2\,1\,1$$
$$\Rightarrow 3\,3\,2\,2 \Rightarrow 4\,3\,3 \Rightarrow 4\,4\,1\,1 \Rightarrow 5\,2\,2\,1 \Rightarrow 3\,3\,2\,1\,1 \Rightarrow 4\,3\,2\,1$$

It takes 13 steps for the counters of 3 4 2 1 to converge to a self-propagating pattern – the "slow motion" or "matching group" a result that is often used to avoid capture of seeds. It is an end of game tactics.

Matching groups can also result ***periodic*** behaviour. Periodicity is a strong principle in signal processing and number theory and automatic control ("limit cycle" or "periodic attractor" in nonlinear dynamics). A 3-period system can be obtained using four counters as:

$$2\,1\,1 \Rightarrow 2\,2 \Rightarrow 3\,1 \Rightarrow 2\,1\,1$$

In this case too, the periodic pattern is obtained by scooping the seeds in the first hole and sowing them in the next holes to the right.

Exercise: What propagating pattern can you obtain from the sequence 2 3 1?

3.4 Magic Numbers

Researchers have used varied approaches to describe and implement African board games. Professor Mayega is noted as the first to implement any of the African games in software. He used Algol 60 to implement Omweso using statistical analysis [3].

Professor Ilukor was the first to use magic numbers in the description of Omweso. He noted the recurrence of the same numbers in the games matrix as he performed mathematical operations on groups of the numbers or seeds in the holes. The observations he made are reproduced here as provided by Wernham. Latter we shall see how his contribution has been used to solve more involving problems in Mankala.

Magic numbers are numeric expressions that occur in statements and expressions. Usually a magic number may not reveal its meaning and intention and therefore should be replaced by a symbolic name. Magic numbers are found many times in matrices and arrays.

When magic numbers are used in games, they are there to disguise intentions and understanding. Your adversary in the game may use them to shield intentions in the next series of moves.

Consider a typical magic number, the so-called magic square. The magic square is an arrangement of the integers from 1 to n^2 such that the sum of the n numbers in each row, in each column, and in the main diagonal and counter diagonal all are the same number, the magic sum **S**. This sum is unique. Here is a typical magic square matrix when n=3

$$Magic(3,3) = \begin{Bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{Bmatrix}$$

Exercises

(a) Write the magic (4,4) square.

(b) Can you prove that $S = \frac{1}{2}n(n^2+1)$?

(c) Is there a magic (2, 2) square matrix for n=2?

Satisfy yourself that magic square matrices can be obtained by adding the same constant value to each element of a magic square matrix.

The 4x4 magic square matrix can result in the game of Omweso and many other African board games. In a magic 4x4 square, the sums of the nonnegative numbers along each row, each column and two main diagonals equal the same sum **s**. We will associate the entries in the magic 4x4 square with the variables $x_1, \ldots, x_{16}$.

Exercises

Write a Java program to find the sums of magic n x n square. The program should print to screen the numbers themselves and their sum.

Show that the product of two orthogonal polynomials gives a result that is a magic number. What is this magic number?

3.4.1 Magic Numbers In Omweso

3.4.1.1 Magic Number 0

The magic number zero results from adding and forming differences using this matrix as an example:

$$\begin{matrix} 16 & (15) & (14) & 13 & 12 & 11 & 10 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & (7) & (6) & 5 & 4 & 3 & 2 & 1 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \end{matrix}$$

The magic number zero is obtained in Omweso when the difference of inner outer diagonal sums to zero. Here is an example. In this example:

$$(15+6)-(14+7)=0$$

3.4.1.2 Magic Number 2

The magic number 2 is obtained under three sum conditions

16	**15**	**(14)**	**13**	**12**	**11**	**10**	**9**
1	**(2)**	**3**	**4**	**5**	**6**	**7**	**8**
8	**7**	**6**	**5**	**4**	**3**	**2**	**1**
9	**10**	**11**	**12**	**13**	**14**	**15**	**16**

a) The differences of outer diagonal sums = 2:

$$([15]+[11])-([14]+[10])=2$$

b) The differences of inner diagonal sums = 2:

$$(5+5)-(4+4)=2$$

c) The differences of one player's diagonal sums =2:

$$(\langle 15\rangle+\langle 3\rangle)-(14+2)=2$$

The numbers used to form the above sums and differences are shown in this matrix.

3.4.1.3 The magic number 8

16	15	14	13	12	**11**	**10**	9
1	(**2**)	3	4	5	6	7	8
8	7	6	5	4	**3**	**2**	1
9	(**10**)	11	12	13	14	15	16

This magic number is obtained under two conditions.

a) The difference of opposite inner and outer row holes =8 and

$$(10-2)=8$$

b) The differences of the diagonal inner and outer products = 8:

$$(10x3)-(11x2)=8$$

3.4.1.4 Magic Number 9

16	15	14	13	12	11	10	9
1	2	(3)	4	5	**6**	**7**	8
8	7	(6)	5	4	**3**	**2**	1
9	10	11	12	13	14	15	16

The magic number 9 is obtained when

a) The sum of inner opposites is formed

$$(6+3)=9$$

b) The differences of inner diagonal multiples is formed

$$(7x3)-(6x2)=9$$

3.4.1.5 Magic Number 16

The magic number 16 is obtained when

c) The difference of inner and outer holes in a column for a player is added to the difference of inner and outer holes in the same column for the second player.

16	(**15**)	14	13	12	11	10	9
1	(**2**)	3	4	5	6	7	8
8	**7**	6	5	4	3	2	1
9	**10**	11	12	13	14	15	16

$$(15-2)+(10-7)=16$$

3.4.1.6 Magic Number 17

16	15	(14)	13	12	11	10	9
1	2	(3)	4	5	6	7	8
8	7	6	5	4	3	**2**	**1**
9	10	11	12	13	14	**15**	**16**

The magic number 17 is formed when

a) The sum of opposite outer row holes is formed and

$$(14+3)=17$$

b) When the differences of outer hole diagonal multiples is formed

$$(2x16)-(1x15)=17$$

3.4.1.7 Magic Number 25

(16)	15	14	13	12	(**11**)	(**10**)	9
1	2	3	4	5	6	7	8
8	7	6	5	4	3	2	1
(9)	10	11	12	13	(**14**)	(**15**)	16

The magic number 25 is formed when

a) The sum of outer opposites is formed and

$$(16+9)=25$$

b) The differences of outer diagonal multiples is formed

$$(11x15)-(10x14)=25$$

3.5 Symmetrical Series In Omweso

In this section we discuss several symmetrical series that arise from the game of Omweso. They include symmetrical square, decreasing integer symmetrical and mirror image even series. These series have two characteristics.

a) In each case a pair of identical series is observed.
b) The symmetry of the series is across the middle vertical dividing line. This line divides the board into two 4x4 matrices as:

$$O(4,8) = L(4,4) : R(4,4)$$

where L(4,4) is the left 4x4 and R(4,4) is the right 4x4 matrices of the Omweso board.

3.5.1 Symmetrical Odd Integer Square Series

A symmetrical square of decreasing odd integers is formed across the board by multiplying differences of holes in columns. The difference of outer holes in each of the columns multiplied by the difference of inner holes in the same column is a square sequence of the odd numbers between 1 and 7.

(16)	15	14	13	12	(**11**)	(**10**)	9
1	2	3	4	5	6	7	8
8	7	6	5	4	3	2	1
(9)	10	11	12	13	(**14**)	(**15**)	16

$$(16-9)\boldsymbol{x}(8-1) = 7^2 = 49$$
$$(15-10)\boldsymbol{x}(7-2) = 5^2 = 25$$
$$(14-11)\boldsymbol{x}(6-3) = 3^2 = 9$$
$$(13-12)\boldsymbol{x}(5-4) = 1^2 = 1$$

This odd square series is repeated for columns 5 to 8. Therefore across the board, the series formed is:

49 25 9 1 1 9 25 49

Considering these integers form the weights of a symmetrical polynomial, the weights are distributed as shown in this Figure.

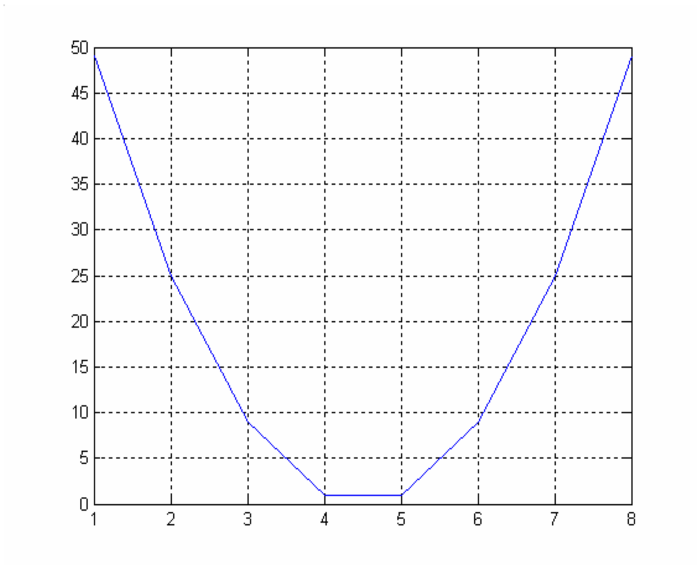


Figure 9: Symmetrical Odd Square Numbers

3.5.2 Decreasing Symmetrical Integer Series

A pair of decreasing integer series is obtained by subtracting the sum of inner holes from the outer holes in each column

(16)	15	14	13	12	(**11**)	(**10**)	9
1	2	3	4	5	**6**	**7**	8
8	7	6	5	4	**3**	**2**	1
(9)	10	11	12	13	(**14**)	(**15**)	16

Top Row, left to right	Bottom Row, left to right
$16-(8+1)=7$	$9-(8+1)=0$
$15-(7+2)=6$	$10-(7+2)=1$
$14-(6+3)=5$	$11-(6+3)=2$
$13-(5+4)=4$	$12-(5+4)=3$
$12-(4+5)=3$	$13-(4+5)=4$
$11-(3+6)=2$	$14-(3+6)=5$
$10-(2+7)=1$	$15-(2+7)=6$
$9-(1+8)=0$	$16-(1+8)=7$

Table 7: Symmetrical Pair of Integer Series

The procedure depicted in this table result to the pair of sequences.

7 6 5 4 3 2 1 0

0 1 2 3 4 5 6 7

3.5.3 Symmetrical Mirror Image Even Series

In this new characteristic of Omweso matrices, a mirror image series arise when products of the inner holes for each column are derived. The resulting series are:

8 14 18 20 20 18 14 8

Observe the mirror image in the middle between the two 20s. These values are derived using the following procedures.

Mirror Image Series
$(1x8) = 8$
$(2x7) = 14$
$(3x6) = 18$
$(4x5) = 20$
$(5x4) = 20$
$(6x3) = 18$
$(7x2) = 14$
$(8x1) = 8$

Table 8: Symmetrical Even Integer Series

Figure 8 is a plot of the amplitudes of the array

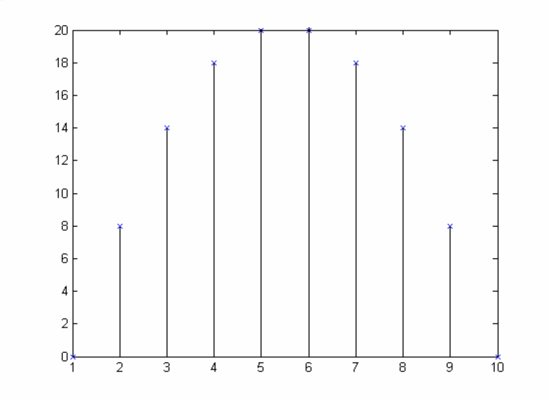


Figure 10: Symmetrical Mirror Image Even Series

3.6 References

[1] Ron Eglas, Recursive Numeric Sequences in Africa, http://www.ccru.net/digithype/recursivenum.htm

[2] Zaslavsky, Claudia. *Africa Counts*. Boston: Prindle, Weber & Schmidt inc. 1973

[3] J.V. Mayega, "Omweso - A Mathematical Investigation of an African Board Game", Dept. of Mathematics, Makerere University, Kampala, Uganda, 1974.

Chapter 4

Never Ending Moves

Introduction

As is normal in many games, the possibility of a recursive game that never ends is a feature of many of the board games. There are two typically never ending moves [1], the so-called *trivial never-ending moves* and the *complex never-ending moves* each with distinct characteristics.

4.1 Trivial Never Ending Moves

Trivial never-ending moves have the following characteristic properties:

- The pattern of repetitions can be seen immediately without experimentation
- The board is left in exactly the same state space after each move with the exception of the position of the starting hole that is rotated
- Triangular numbers are involved with patterns of 4, 3, 2, 1 appearing
- When the pattern is divided up into four 2x2 matrices, a 2x2 identity matrix and certain matrices appear repeatedly or in rotated form

Consider a typical trivial never-ending move is shown as shown in the following matrix.

$$\boldsymbol{T}_3 = \begin{matrix} 2 & 1 & 0 & 2 & 1 & 0 & 2 & 1 \\ 3 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \end{matrix}$$

Represent T_3 as four 2x2 matrices in the form by concatenating them in the order of appearance in the original full matrix as:

$$T_3 = T_1 \vdots T_2 \vdots I_2 \vdots T_{2r}$$

where, T_{2r} is T_2 matrix rotated first around a horizontal axis, then around a vertical axis and $\vdots$ represents concatenation of matrices. This matrix property has not been reported in any previous writings. In this representation

$$I_2 = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}; T_1 = \begin{matrix} 2 & 1 \\ 3 & 0 \end{matrix}; T_2 = \begin{matrix} 0 & 2 \\ 1 & 2 \end{matrix}$$

When T_2 is rotated around a horizontal axis we have: $T_{rh} = \begin{matrix} 1 & 2 \\ 0 & 2 \end{matrix}$ and when this resulting matrix is rotated around a vertical axis we have the result:

$$T_{2r} = \begin{matrix} 2 & 1 \\ 2 & 0 \end{matrix}.$$

4.2 Complex Never Ending Moves

When moves in a game lead to infinite looping, it is called complex never-ending moves. A typical example is the so-called “Hudson’s 32” which is a 61, 776 iteration never-ending game. Wernham [1] reports that two players working in 4 hours relays in Kampala failed to find an end-point or repetition in the game.

The complex positions have the following characteristic properties:

- Large numbers of repetitions usually more than 200 before the positions re-appears again
- The position of the starting hole is the same when the seeds re-appears again in the same sequence
- The number of iterations is always divisible by 4
- There are no obvious patterns occurring although we can now define each matrix by a concatenation of four 2x2 matrices and in some cases, the same 2x2 matrix re-occurs.

4.2.1 Kyagaba-32

Kyagaba-32 complex never-ending moves (61, 776 iterations) matrix uses 32 seeds:

$$\boldsymbol{K}_{32} = \begin{matrix} 3 & 2 & 3 & 2 & 3 & 2 & 3 & 1 \\ 2 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \end{matrix}$$

Let

$$\boldsymbol{K}_{32} = \boldsymbol{K}_1 \vdots \boldsymbol{K}_2 \vdots \boldsymbol{K}_2 \vdots \boldsymbol{K}_3$$

be a concatenation of four 2x2 matrices so that

$$\boldsymbol{K}_1 = \begin{matrix} 3 & 2 \\ 2 & 2 \end{matrix}; \boldsymbol{K}_2 = \begin{matrix} 3 & 2 \\ 1 & 2 \end{matrix}; \boldsymbol{K}_3 = \begin{matrix} 3 & 1 \\ 1 & 2 \end{matrix}.$$

Wernham and Ilukor gave other complex never-ending moves. There is also one by Hudson-32 reported by Wernham which is identical with the Kyagaba-32.

4.2.2 Ilukor-25 Case 1

The first case of Ilukor-25 complex never-ending moves (852 iterations) matrix consisting of 25 seeds is:

$$\boldsymbol{L}_{25} = \begin{matrix} 1 & 2 & 1 & 0 & 1 & 4 & 1 & 0 \\ 10 & 0 & 3 & 0 & 1 & 0 & 0 & 1 \end{matrix}$$

Representing it as a concatenation of four matrices we have:

$$\boldsymbol{L}_{25} = \boldsymbol{L}_1 \vdots \boldsymbol{L}_2 \vdots \boldsymbol{L}_3 \vdots \boldsymbol{I}_2$$

so that we have three distinct 2x2 matrices and a 2x2 identity matrix involved. They are

$$\boldsymbol{L}_1 = \begin{matrix} 1 & 2 \\ 10 & 0 \end{matrix}; \boldsymbol{L}_2 = \begin{matrix} 1 & 0 \\ 3 & 0 \end{matrix}; \boldsymbol{L}_3 = \begin{matrix} 1 & 4 \\ 1 & 0 \end{matrix}; \boldsymbol{I}_2 = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

4.2.3 Ilukor-25 Case 4

The following matrix represents another case of Ilukor-25 complex never-ending moves (198, 288 iterations).

$$\hat{\boldsymbol{L}}_{25} = \begin{matrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 17 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{matrix}$$

This matrix has a simple format as

$$\hat{\boldsymbol{L}}_{25} = \boldsymbol{L}_4 \vdots \boldsymbol{I}_2 \vdots \boldsymbol{I}_2 \vdots \boldsymbol{I}_2$$

where

$$\boldsymbol{L}_4 = \begin{matrix} 1 & 0 \\ 17 & 1 \end{matrix}$$

4.2.4 Wernham-28

The Wernham-28 complex never-ending moves (264 iterations) matrix is a slight modification of the Ilukor-25 and uses three more seeds in the form:

$$\boldsymbol{W}_{28} = \begin{matrix} 1 & 2 & 1 & 0 & 1 & 4 & 3 & 1 \\ 10 & 0 & 3 & 0 & 1 & 0 & 1 & 0 \end{matrix}$$

Representing it as a concatenation of four matrices we have:

$$\boldsymbol{W}_{28} = \boldsymbol{L}_1 \vdots \boldsymbol{L}_2 \vdots \boldsymbol{L}_3 \vdots \boldsymbol{W}_1$$

and

$$\boldsymbol{W}_1 = \begin{matrix} 3 & 1 \\ 1 & 0 \end{matrix}$$

4.2.5 Jonkers, Uiterwijk and de Voogt-40

The Jonkers-40 (Uiterwijk-40 or Voogt-40) complex never-ending move matrix is given as

$$\boldsymbol{J}_{40} = \begin{matrix} 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 \\ 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \end{matrix}$$

The concatenated matrix form is

$$\boldsymbol{J}_{40} = \boldsymbol{J}_1 \vdots \boldsymbol{J}_1 \vdots \boldsymbol{J}_1 \vdots \boldsymbol{J}_1$$

where

$$\boldsymbol{J}_1 = \begin{matrix} 3 & 2 \\ 2 & 3 \end{matrix}$$

4.3 A Theorem of Endless Moves

Professor Steven P. Meyer [2] provided this theorem in 2001 and reported by Wernham in his paper [1]. We are reproducing the theorem in its original form with a modification in the indexing to facilitate reading and understanding. Most of the notation used in this chapter is that of Meyer.

4.3.1 Notation

On a 16-hole Mankala board, let the hole from which the next iteration is to begin be “0” and number the remaining holes in a clockwise direction from 1 to 15. Let y(k) denote the number of seeds in hole k. This proof given below can be extended for an ‘k’ hole Mankala board, with the test and proof working to mod k+1.

4.3.2 Theorem of Never Ending Moves

A move in 16-hole Mankala is endless if it satisfies all of the following incongruencies for all values of i and j, with i running from 0 to 15 and j running from 0 to 15 - i:

y(i) not congruent to i + 1 (mod 17)	**A**
y(i) not congruent to i - 1 (mod 17)	**B**
y(i+j) not congruent to y(i) + j + 1 (mod 17)	**C**
y(i+j) not congruent to y(i) + j – 1 (mod 17)	**D**

4.3.2.1 Outline of Proof

At the start of one iteration, assume A, B, C and D to be true for all permissible values of i and j. We will prove that they are true for all permissible valus of i and j at the start of the next iteration. Therefore they will hold for all iterations. Then incongruency A for the case i = 0 implies that the number of seeds in hole 0 can never be congruent to 1 (and therefore can never equal 1). We will establish that all of the incongruencies hold by assuming the negation of each one and showing that in each case it implies the negation of an incongruency in the previous iteration, which contradicts our assumption.

4.3.2.2 Notation

Let A(i) refer to incongruency A for a particular value of i.
Let B(i) refer to incongruency B for a particular value of i.
Let C(i,j) refer to incongruency C for particular values of i and j.
Let D(i,j) refer to incongruency D for particular values of i and j.
Let x(k) denote the number of seeds in hole k at the start of iteration 1.
Let y(k) denote the number of seeds in hole k at the start of iteration 2.

Let ~ mean "is congruent modulo 17".

For example, if the arrangement of seeds is as follows:

12	5	6	0	2	11	9	0
2	6	3	3	5	0	1	8

and if the move begins by lifting the twelve seeds from the upper left-hand hole, then y(0) = 12, y(1)=5, y(2) = 6, y(3) = 0, y(4) = 2, y(5) = 11, y(6) = 9, y(7) =0, y(8) = 8, y(9) = 1, y(10) = 0, y(111) = 5, y(12) = 3, y(13) = 3, y(14) = 6, y(15) = 2

4.3.2.3 Case 1: Suppose x(0) is a multiple of 16

Then x(0) = 16n for some n. Thus

x(0) ~ -n (1)
y(0) = n (2)
y(k) = x(k) + n for k > 0 (3)

Suppose A(0) is violated on iteration 2. Then

y(0) ~ 1
n ~ 1 by (2)
x(0) ~ -1 by (1)
contradicts B(0) on iteration 1

Suppose A(i) is violated on iteration 2, i > 0. Then

y(i) ~ i + 1
x(i) + n ~ i + 1 by (3)
x(i) ~ i + 1 – n
x(0) ~ x(0) + i + 1 by (1)
contradicts C(0,i) on iteration 1

Suppose B(0) is violated on iteration 1. Then

y(0) ~ -1
n ~ -1 by (2)
x(0) ~ 1 by (1)
contradicts A(0) on iteration 1

Suppose B(i) is violated on iteration 2, i > 0. Then

y(i) ~ i -1

$x(i) + n \sim i - 1$ by (3)
$x(i) \sim i - 1 - n$
$x(i) \sim x(0) + i - 1$ by (1)
contradicts D(0,i) on iteration 1

Suppose C(0,0) is violated on iteration 2. This can never happen, since C(0,0) implies $0 \sim 1$.

Suppose C(0,j) is violated on iteration 2, j > 0. Then
$y(j) \sim y(i) + j + 1$
$x(j) + n \sim n + j + 1$ by (3) and (2)
$x(j) \sim j + 1$
contradicts A(j) on iteration 1

Suppose C(i, 0) is violated. This can't happen since it implies $y(i) \sim y(i) + 1$

Suppose C(i,j) is violated on iteration 2, i > 0 , j> 0. Then
$y(i+j) \sim y(i) + j + 1$
$x(i+j) + n \sim x(i) + n + j + 1$ by (3)
$x(i+j) \sim x(i) + j + 1$
contradicts C(i,j) on iteration 1
D(0,0) can't happen.

Suppose D(0,j) is violated on iteration 2 with j > 0. Then
$y(j) \sim y(0) + j - 1$
$x(j) + n \sim n + j - 1$ by (3) and (2)
$x(j) \sim j - 1$
contradicts B(j) on iteration 1
D(i,0) can't happen.

Suppose D(i,j) is violated on iteration 2, with i > 0 and j > 0. Then
$y(i+j) \sim y(i) + j - 1$
$x(i+j) + n \sim x(i) + n + j - 1$ by (3)
$x(i+j) \sim x(i) + j - 1$
contradicts D(i,j) on iteration 1

4.3.2.4 Case 2: x(0) is not a multiple of 16

Then $x(0) = 16n + m$ for some n and m, where $0 < m < 16$. Thus
$x(0) \sim m - n$ (4)
$y(0) = x(16-m) + n + 1$ (5)
$y(m) = n$ (6)
If $0 < k < m$ then $y(k) = x(16-m+k) + n + 1$ (7)
If $k > m$ then $y(k) = x(k-m) + n$ (8)

To save space we use boldface type to denote incongruencies that are hypothetically violated on iteration 2. We will use normal type to denote incongruencies that are consequently violated on iteration 1. We will omit cases where j = 0, since these are all vacuous.

A(0):
$y(0) \sim 1$
$x(16-m) + n + 1 \sim 1$ by (5)
$x(16-m) \sim -n$
$x(16-m) \sim x(0) - m$ by (4)
$x(16-m) \sim x(0) + (16-m) + 1$ because $16 \sim -1$

C(0,16-m)

A(i), 0 < i < m:
y(i) ~ i + 1
x(16-m+i) + n + 1 ~ i + 1 by (5)
x(16-m+i) ~ i – n
x(16-m+i) ~ x(0) + i – m by (4)
x(16-m+i) ~ x(0) + (16-m+i) + 1
C(0,16-m+i)

A(m):
y(m) ~ m + 1
n ~ m + 1 by (6)
-1 ~ m – n
-1 ~ x(0) by (4)
B(0)

A(i), i > m :
y(i) ~ i + 1
x(i-m) + n ~ i + 1 by (8)
x(i-m) ~ (i-m) + m – n + 1
x(i-m) ~ (i-m) + x(0) + 1 by (4)
C(0, i-m)

B(0):
y(0) ~ -1
x(16-m) + n + 1 ~ -1 by (5)
x(16-m) ~ (16 – m) + m – n – 1 because 16 ~ -1
x(16-m) ~ (16 – m) + x(0) - 1 by (4)
D(0, 16-m)

B(i), 0 < i < m :
y(i) ~ i – 1
x(16-m+i) + n + 1 ~ i - 1 by (7)
x(16-m+i) ~ (16 – m + i) + m – n – 1
x(16-m+i) ~ (16 – m + i) + x(0) – 1 by (4)
D(0, 16-m+i)

B(m):
y(m) ~ m – 1
n ~ m – 1 by (6)
1 ~ m – n
1 ~ x(0) by (4)
A(0)

B(i), i > m :
y(i) ~ i – 1
x(i-m) + n ~ i – 1 by (8)
x(i-m) ~ (i – m) + m – n – 1
x(i-m) ~ (i – m) + x(0) – 1 by (4)
D(0, i-m)

C(0, j), 0 < j < m :
y(j) ~ y(0) + j + 1
x(16-m+j) + n + 1 ~ x(16-m) + n + 1 + j + 1 by (7) and (5)

x(16-m+j) ~ x(16-m) + j + 1
C(16-m, j)

C(0, m):
y(m) ~ y(0) + m + 1
n ~ x(16-m) + n + 1 + m + 1 by (6) and (5)
x(16-m) ~ (16 – m) – 1
B(16 – m)

C(0, j), j > m :
y(j) ~ y(0) + j + 1
x(j-m) + n ~ x(16-m) + n + 1 + j + 1 by (8) and (5)
x((j-m) + (16-j)) ~ x(j-m) + (16 – j) – 1
D(j-m, 16-j)

C(i, j), 0 < i < m , 0 < j < m – i :
y(i+j) ~ y(i) + j + 1
x(16-m+i+j) + n + 1 ~ x(16-m+i) + n + 1 + j + 1 by (7)
x(16-m+i+j) ~ x(16-m+i) + j + 1
C(16-m+i, j)

C(i, m-i), 0 < i < m :
y(i+m-i) ~ y(i) + m – i + 1
y(m) ~ y(i) + m – i + 1
n ~ x(16-m+i) + n + 1 + m – i + 1 by (6) and (7)
x(16-m+i) ~ (16 – m + i) – 1
B(16-m+i)

C(i, j), 0 < i < m, m – i < j < 16 – i :
y(i+j) ~ y(i) + j + 1
x(i+j-m) + n ~ x(16-m+i) + n + 1 + j + 1 by (8) and (7)
x((i+j-m) + (16-j)) ~ x(i+j-m) + (16 – j) – 1
D(i+j-m, 16-j)

C(m, j), j > 0 :
y(m+j) ~ y_m + j + 1
x(m+j-m) + n ~ n + j + 1 by (8) and (6)
x(j) ~ j + 1
A(j)

C(i, j), i > m , j > 0 :
y(i+j) ~ y(i) + j + 1
x(i+j-m) + n ~ x(i-m) + n + j + 1 by (8)
x(i+j-m) ~ x(i-m) + j + 1
C(i-m, j)

D(0, j) , 0 < j < m :
y(j) ~ y(0) + j – 1
x(16-m+j) + n + 1 ~ x(16-m) + n + 1 + j – 1 by (7) and (5)
D(16-m, j)

D(0, m):
y(m) ~ y(0) + m – 1
n ~ x(16-m) + n + 1 + m – 1 by (6) and (5)
x(16-m) ~ (16 – m) + 1
A(16-m)

D(0, j), j > m :
y(j) ~ y(0) + j – 1
x(j-m) + n ~ x(16-m) + n + 1 + j – 1 by (8) and (5)
x((j-m) + (16-j)) ~ x(j-m) + 1 + 16 – j
D(j-m, 16-j)

D(i, j), 0 < i < m , 0 < j < m – i :
y(i+j) ~ y(i) + j – 1
x(16-m+i+j) + n + 1 ~ x(16-m+i) + n + 1 + j – 1 by (7)
D(16-m+i, j)

D(i, m-i) , 0 < i < m :
y(m) ~ y(i) + m – i – 1
n ~ x(16-m+i) + n + 1 + m – i – 1 by (6) and (7)
x(16-m+i) ~ 16 – m + i + 1
A(16 – m + i)

D(i, j), 0 < i < m , m – i < j < 16 – i :
y(i+j) ~ y(i) + j – 1
x(i+j-m) + n ~ x(16-m+i) + n + 1 + j – 1 by (8) and (7)
x((i+j-m) + (16-j)) ~ x(i+j-m) + (16 – j) + 1
C(i+j-m, 16-j)

D(m, j), j > 0 :
y(m+j) ~ y_m + j – 1
x(m+j-m) + n ~ n + j – 1 by (8) and (6)
x(j) ~ j – 1
B(j)

D(i, j), i > m , j > 0 :
y(i+j) ~ y(i) + j – 1
x(i+j-m) + n ~ x(i-m) + n + j – 1 by (8)
D(i-m, j)

4.4 Meyer Test of Incongruencies

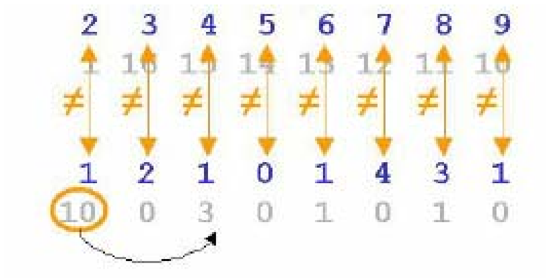
Wernham [1] carried out a test of the Meyer theorem to establish the incongruences in statements A, B, C and D of the theorem. Those tests are reproduced here.

4.4.1 Meyer Test A

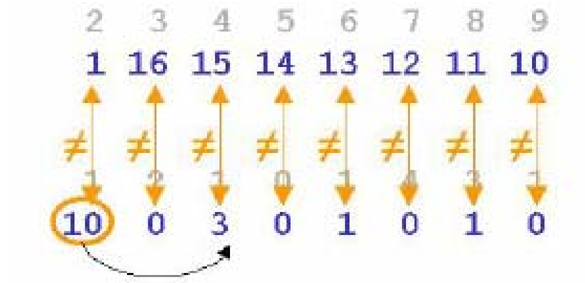
This test establishes that the number of seeds in hole $i \neq i+1$. The count i is clockwise in the matrix:

$$
\begin{array}{cccccccc}
i & \rightarrow & & & & & & \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
0 & 15 & 14 & 13 & 12 & 11 & 10 & 9
\end{array}
$$

The test is based on the $\mathbf{W_{28}}$ matrix the 264 iterations case. The diagrams below are reproduced from Wernham's paper.
Checking the top row with i+1:

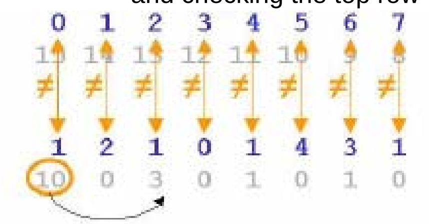


Checking the bottom row with i+1:

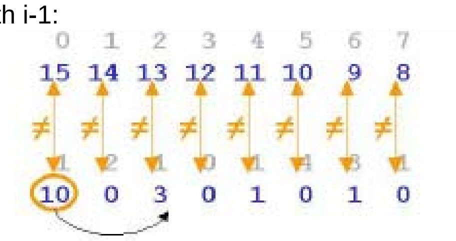


4.4.2 Meyer Test B

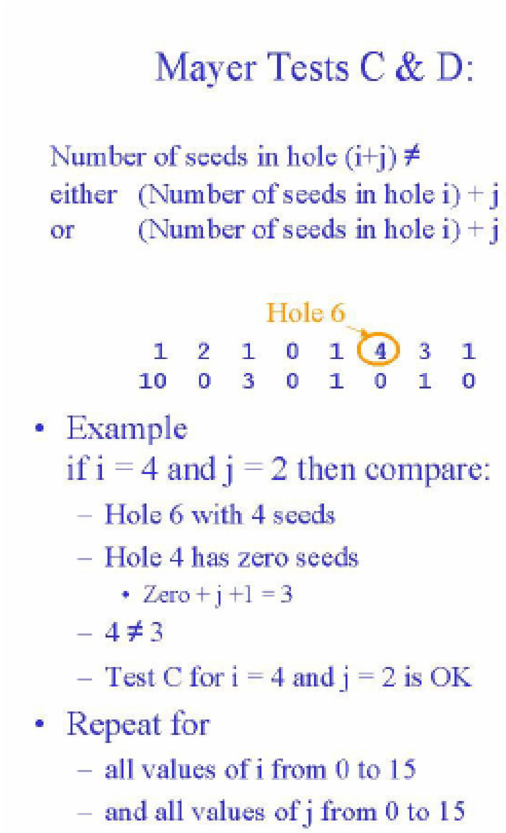
The number of seeds in hole, $i \neq i-1$ and checking the top row with i-1:



Checking the bottom row with i-1:



4.4.3 Meyer Test C and D



Meyer's proof is reproduced in this chapter. The proof is iterative, first proving that Test A is violated ad the end of a sowing process if the last seed is dropped in an empty hole. There after he then proved that conditions C, D and E support condition A.

- Test A is violated at the end of a move if Test B, C or D are violated at the beginning of the move
- Test B is violated only if Test A or D were violated previously
- Tests C and D can only be violated if Tests A, B, C or D were violated previously.

4.5 References

[1] Brian Wernham, "Omweso: The Royal Mankala Game of Uganda – A General Overview of Current Research", International Omweso Society, http://www.omweso.org, 2002, pp. 1-26.

[2] S. Meyer, Theorem of Never Ending Moves, Dept. of Physics, Milwaukee School of Engineering, Milwaukee, Wisconsin, USA, 2001

Chapter 5

Diagonal Count Arrays

5 Introduction

There is a second proof of never ending moves tucked away in the Internet. John Earl's [1] proof is provided in this chapter. We have left the steps in tact but modified the variables to be in line with the style of the book. The rest of the chapter is core John Earl's work.

5.1 Diagonal Count Arrays

Steven Mayer's set of incongruences, which imply that certain positions represent never ending moves, admits a geometric interpretation, which we describe here in terms of diagonal count arrays provides an alternative proof of Mayer's Theorem, as well as leading to some further results concerning the number of seeds sown.

For an *n*-hole multiple-lap mankala position we adopt the notation
$\{y\} = \{y(0) \quad y(1) \quad \cdots \quad y(k-1) \quad (y(k)) \quad \cdots \quad y(n-1)\}$.
The *y(j)* are the numbers of seeds in the *n* holes, in the direction of play, with the active cell in parentheses and *y(n)* identified with *y(0)*. We consider only the case where capture is not possible; the problem of interest is deciding whether or not a move is endless.

A move may ***start*** whenever *y(k)* >1 and consists of lifting the *y(k)* seeds and placing them, one at a time, in the next *y(k)* holes, ending with hole *j* = (*k*+ *y(k)*) mod *n*, to give a new position $\{x\} = \{x(0) \quad x(1) \quad \cdots \quad x(j-1) \quad (x(j)) \quad x(j+1) \quad \cdots \quad x(n-1)\}$.
Note that in this situation *x(j-1)* > 0 and *x(j)* ≥ 1 . If *x(j)* > 1 then we repeat the same steps as before, otherwise the move *terminates*. Starting with {*x*} it is possible to work backwards to determine its immediate predecessor {*y*}. Simply remove one seed from each cell, starting with *x(j)* and move in the reverse direction, until an empty cell is reached and place the accumulated seeds in the empty cell. Provided *y(k-1)* ≠ 0, we can also find the unique immediate predecessor of {*y*}.

If *y(k-1)* = 0 then {*y*} has no predecessor and is a *maximal* start position.
For a position $\{u\} = \{u(0) \quad u(1) \quad u(2) \quad \cdots \quad (u(k)) \quad \cdots \quad u(n-1)\}$ we form the corresponding *diagonal count array* $(d(0) \quad d(1) \quad d(2) \quad \cdots \quad d(n))$ with *n* + 1 elements as follows:

(i) Rotate the position {*u*}so that active cell is in last place, to give $\{y(0) \quad y(1) \quad y(2) \quad \cdots \quad (y(n-1))\}$, with $y(j) = u(i)$, where $i = (k+1+j)\bmod(n)$;

(ii) For $0 \le j \le n$ set *d(j)* = number of points $(i, y(i))$ lying on the *j*-diagonal $x + y \equiv j \bmod (n+1)$, i.e. having $i + y(i) \equiv j \bmod (n+1)$;

(iii) Add 1 to *d(n-1).*

N.B. $\sum d(j) = n + 1$.

Proposition*:* If we start with a position {*u*} and perform one iteration to give {*v*} then the diagonal count array for {*v*} is the same as that for {*u*} rotated to the left by *u(k)* places.

Proof: We consider a start position $\{y(0) \quad y(1) \quad y(2) \quad y(3) \quad \cdots \quad \cdots \quad (y(n-1))\}$ where $y(n-1) = mn + p \ge 2, 0 \le p < n$.
The seeds in cell *n* – 1 are lifted and sowed in turn one at a time into the cells, starting with y(0), thus

- increasing *y(i)* to $y(i) + m + 1, 0 \le i \le p - 1$;
- increasing *y(i)* to $y(i) + m, p \le i c n - 2$;
- decreasing *y(n-1)* to *m*

Rotation to the right by *n – p* places moves new active cell *y(p-1)* to last place $y¢_{n-1}$ and new *y(n-1)* to *y*′(n-p-1) . For 0 ≤ *x* ≤ *p* – 1,

$x' = x + n - p;\ y' = y + m + 1$, so that
$x' + y' = x + y + n + 1 - p + m \equiv j - p + m + n + 1 \bmod (n+1) \equiv j - p + m \bmod (n+1)$.

For $p \le x \le n - 2$,
$x' = x - p, y' = y + m,$ so that
$x' + y' \equiv x + y - p + m \equiv j - p + m.$
For these *n*–1 points in total, the original adds to *j* count ⇔ its image adds to the (*j*–*p*+ *m*) count. Since $mn = m(n+1) - m \equiv -m \bmod (n+1)$ this is equivalent to rotation to the left by *p* + *mn.*
The remaining original point (*n*-1, *mn* + *p*) added one to the (*n*-1+ *p*-*m*) count. The rotation to the left by *p* – *m* of the diagonal count array shifts the effect of this increment to the *n*-1 term and is equivalent to the addition of one to the *n*-1 diagonal count following iteration. The image point (*n*-1-*p*, *m*) increases the (*n*-1-*p*+*m*) diagonal count by one; the latter increase is equivalent to the one which is added to the *n*-1 entry of the original array, rotated left by *p*-*m*.

Corollary 1: If a chain of iterations starts with {*u*} and ends with {*v*} then the diagonal count array for {*v*} is the same as that for {*u*} rotated to the left by the total number of seeds sown.
Corollary 2: If, in Corollary 1, {*v*} is a rotation of {*u*} then their diagonal count arrays will be the same and, hence, a rotation of the diagonal count array by the number of seeds sown will leave it unchanged.

Corollary 3: If, in Corollary 2, n+1 is prime then the rotation of the diagonal count arrays will have to be *complete* rotations unless the *diagonal count array* = {1} = (1 1 1 1 … 1). It follows that the number of sowings will have to be a multiple of n+1 unless the *diagonal count array* = {1}.

The Omweso positions

$$\{0 \quad 1 \quad 2 \quad 0 \quad 1 \quad 2 \quad 0 \quad 1 \quad 2 \quad 0 \quad 1 \quad 2 \quad 0 \quad 1 \quad 2 \quad (3)\},$$

$$\{0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad (5)\} \text{ and}$$

$$\{0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12 \quad 13 \quad 14 \quad (15)\}$$

which all produce a rotation after one iteration must have *diagonal count array*s = {1} as is easily verified.

Corollary 4: *Mayer's Theorem*
If the non-zero diagonal count array entries are separated from each other by zero entries then any associated position corresponds to a never-ending move.

Proof: If the position terminates after a finite number of iterations it is an iteration of a position in which the active cell is preceded by an empty cell. The position $\{y(0) \quad y(1) \quad y(2) \quad \cdots \quad y(k) \quad \cdots \quad 0 \quad (y(n-1))\}$ has *d(n-2)* > 0 (from point (*n*-2,0)) and *d(n-1)* > 0 (always). Any rotation of the diagonal count array has at least two adjacent non-zero entries, unless they were to become first and last places, when pair *d(n-1)* and *d(n)* would be non-zero.

An *orbit* is formed by a position together with all its iterations and predecessors. The total number of seeds in a position and the set of rotations of the diagonal count array are *invariants* over the orbit. Other examples would be of interest.

Example

enter * to mark active hole in **{u}**									*									9		
x	0	1	2	3	4	5	6	7	8	9	10	11	12	**13**	14	15				
enter u(x)	2	1	3	1	0	3	4	1	13	0	0	0	1	2	1	3		Position {u}	35	
y(x)	0	0	0	1	2	1	3	2	1	3	1	0	3	4	1	**13**		**Rotate**, last hole active	m=	0
x + y(x)	0	1	2	4	6	6	9	9	9	12	11	11	15	17	15	28			p=	13
j = (x + y(x)) mod 17	0	1	2	4	6	6	9	9	9	12	11	11	15	0	15	11		0 <= j <= 16		
j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
Frequency of j in above	2	1	1	0	1	0	2	0	0	3	0	3	1	0	0	2	0		16	
Add one to entry j = 15	**2**	**1**	**1**	**0**	**1**	**0**	**2**	**0**	**0**	**3**	**0**	**3**	**1**	**0**	**0**	**3**	**0**	**Original DCA**	**17**	
													*							
1 iteration of {y}	1	1	1	2	3	2	4	3	2	4	2	1	**4**	4	1	0		sow 13 to get this	35	
{y'} = Rotate so that active hole is last	4	1	0	1	1	1	2	3	2	4	3	2	4	2	1	**4**		a	35	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		b		

New diagonals	4	2	2	4	5	6	8	10	10	13	13	13	16	15	15	2		(a+b) mod (17)		
j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
Frequency of j for iterated position	0	0	3	0	2	1	1	0	1	0	2	0	0	3	0	2	1		16	
Add one to entry j = 15	**0**	**0**	**3**	**0**	**2**	**1**	**1**	**0**	**1**	**0**	**2**	**0**	**0**	**3**	**0**	**3**	**1**	**DCA of iteration**	**17**	
																			0	
Rotate original DCA by p-m to left	**0**	**0**	**3**	**0**	**2**	**1**	**1**	**0**	**1**	**0**	**2**	**0**	**0**	**3**	**0**	**3**	**1**	**Rotation of original DCA**	**17**	
N.B. Same as rot by -(p+16m)																				
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Difference		
	18	17	15	14	13	12	11	10	9	8	7	6	5	4	3	2				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	17					
33	5	6	7	8	9	10	11	12	13	14	15	16	0	1	2	3	4			
32	4	5	6	7	8	9	10	11	12	13	14	15	16	0	1	2	3			
31	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0	1	2			
30	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0	1			
29	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0			
28	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
27	16	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
26	**15**	16	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14			
25	14	**15**	16	0	1	2	3	4	5	6	7	8	9	10	11	12	13			
24	13	14	**15**	16	0	1	2	3	4	5	6	7	8	9	10	11	12			
23	12	13	14	**15**	16	0	1	2	3	4	5	6	7	8	9	10	11			
22	11	12	13	14	**15**	16	0	1	2	3	4	5	6	7	8	9	10			
21	10	11	12	13	14	**15**	16	0	1	2	3	4	5	6	7	8	9			
20	9	10	11	12	13	14	**15**	16	0	1	2	3	4	5	6	7	8			
19	8	9	10	11	12	13	14	**15**	16	0	1	2	3	4	5	6	7			
18	7	8	9	10	11	12	13	14	**15**	16	0	1	2	3	4	5	6			
17	6	7	8	9	10	11	12	13	14	**15**	16	0	1	2	3	4	5			
16	5	6	7	8	9	10	11	12	13	14	**15**	16	0	1	2	3	4			
15	4	5	6	7	8	9	10	11	12	13	14	**15**	16	0	1	2	3			
14	3	4	5	6	7	8	9	10	11	12	13	14	**15**	16	0	1	2			
13	2	3	4	5	6	7	8	9	10	11	12	13	14	**15**	16	0	1			
12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**15**	16	0			
11	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**15**	16			
10	16	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**15**			
9	15	16	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14			
8	14	15	16	0	1	2	3	4	5	6	7	8	9	10	11	12	13			
7	13	14	15	16	0	1	2	3	4	5	6	7	8	9	10	11	12			
6	12	13	14	15	16	0	1	2	3	4	5	6	7	8	9	10	11			
5	11	12	13	14	15	16	0	1	2	3	4	5	6	7	8	9	10			
4	10	11	12	13	14	15	16	0	1	2	3	4	5	6	7	8	9			
3	9	10	11	12	13	14	15	16	0	1	2	3	4	5	6	7	8			
2	8	9	10	11	12	13	14	15	16	0	1	2	3	4	5	6	7			

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

5.2 References

[1] John Earl, "Diagonal count arrays in *n*-hole Mancala" CT4 6PU, U.K. John.Earl@u(k)gateway.net

Chapter Six

Game Defining Matrices

6 Introduction

We have used 2x2 matrices to describe the state space and state of events on the game board. It appears 2x2 matrices that can be used to define never ending moves and starting moves form complete sets. This means the same matrices could be used to define new never ending moves and also start up state spaces. In this section we discuss these matrices further. Defining matrices possess four characteristics:

- m of such matrices must be concatenated to form a game state and m is the number of holes per row of the board divided by 2
- the sum of the seeds of those m matrices should be N where N is the number of seeds allowed for the game state space. For example, in Omweso at set up state space, N=32
- if the sum of elements of any of the 2x2 matrices is odd, it must be paired with another defining matrix whose sum of elements is also odd
- the set of defining matrices to set up for one player is always a rotated version of the same matrices for the opponent

6.1 Matrices Defining Never Ending Moves

In Chapter 4 we derived a group of matrices that have been used to define never ending moves. Are these matrices unique? If so how many are they? What characteristics do these matrices posses? For examples, are there unique matrices that lead to never ending moves? It appears there are. We have defined eleven (11) matrices as listed in Table 7.

Name	Symbol	Matrix
Identity	I_2	1 0 0 1
Ilukor_1	L_1	1 2 10 0
Ilukor_2	L_2	1 0 3 0
Ilukor_3	L_3	1 4 1 0
Ilukor_4	L_4	1 0 17 1
Jonkers_1	J_1	3 2 2 3
Kyagaba_1	K_1	3 2 2 2
Kyagaba_2	K_2	3 2 1 2

Kyagaba_3	K_3	3 1 1 2
Wernham_1	W_1	3 1 1 0
Trivial_1	T_1	2 1 3 0
Trivial_2	T_2	0 2 1 2

Table 9: Defining Matrices

The matrices in Table 7 have so far been used to define never ending moves. We have not as yet proved that this set of defining matrices is complete for defining never ending moves. There remains an area of research to use groups of four matrices taken from this set to define never ending moves.

6.2 Game Set Up Defining Matrices

Initial game startup matrices are perhaps one of the most important sets of matrices in board games particularly when such games are offered in software. They enable unencumbered optimal selection of startup states particularly if those matrices are known to posses optimal defense characteristics. Following the work in [1] using the so-called "junior grouping" and "senior grouping" one can define efficient 2x2 matrices that define efficient set up states.

6.2.1Junior Group Defining Matrices

The following matrices represent a snapshot of the conditions that exist in a junior start up. An exclusive set is not the intention rather a collection of matrices that facilitate the setting up of the game of Omweso for none experts.

Name	Symbol	Matrix
Identity six	I_6	6 0 0 6
Null	N_0	0 0 0 0
Start up 1	S_1	0 0 7 3
Start up 2	S_2	0 0 3 3
Start up 3	S_3	5 0 0 0
Start up 4	S_4	7 2 0 0
Start up 5	S_5	4 2 0 0

Table 10: Junior Defining Matrices

6.2.2 Senior Group Defining Matrices

Senior grouping matrices include at least one hole with more seeds and usually up to 23 seeds in some cases.

Name	Symbol	Matrix
Setup 6	S_6	23 0; 0 3
Setup 7	S_7	3 0; 0 20
Setup 8	S_8	3 19; 0 0
Setup 9	S_9	0 2; 2 0
Setup 10	S_{10}	1 1; 0 0
Setup 11	S_{11}	2 3; 0 0
Setup 12	S_{12}	3 18; 0 0
Setup 13	S_{13}	0 2; 0 2
Setup 14	S_{14}	3 17; 0 0
Setup 15	S_{15}	3 0; 0 17
Setup 17	S_{17}	0 3; 0 0
Setup 18	S_{18}	4 4; 0 0
Setup 19	S_{19}	0 0; 3 7 = S_{1r}
Setup 20	S_{20}	3 0; 0 23 = S_{6r}
Setup 21	S_{21}	0 0; 4 4 = S_{18r}

Table 11: Senior Defining Matrices

The matrices S_{19}, S_{20} and S_{21} are rotated versions of S_1, S_6 and S_{18} respectively. All the matrices have at least two elements that are zero. Only six of them have sums of elements less than 10. The sum of most is more than 20.

6.2.3 Game Set up with Defining Matrices

All board games can be set up with defining matrices of the right types. Rules of the game and the experience of the player dictate the right types. In a software based game, set up with pre-determined matrices places some control in the hands on the human player to pre-determine his own set up without it being forced on him by the computer software.

Four matrices must be concatenated and used for the set up. In a game of Omweso, the sum of the seeds in the matrices must add up to 32.

6.3 References

[1] Brian Wernham, “Omweso: The Royal Mankala Game of Uganda – A General Overview of Current Research”, International Omweso Society, http://www.omweso.org, 2002, pp. 1-26.

Part Two

Strategies and Research

Chapter 7

Research Problems in Bao

7 Introduction

Technical knowledge of African board games is still in its infancy stage. Research into them has progressed a bit within the last decade with the most recent being the solution of the Warri game reported in the IEEE [1] and [4]. Warri rules and software design and testing are well covered in [4]. In this Chapter we pursue some of the existing research questions. David Mackay [2] asked whether "a move in a game of Bao" can "end up in an infinite loop?" and provided some answers. This chapter is a summary of those answers.

7.1 Kutakata [18]

In a type of move called **kutakata**, involving relay sowing, a player selects any shimo containing at least two kete (but not more than 15). The player picks up all the kete from it, then sows either anti-clockwise or clockwise around their two rows. One kete is deposited in each of the mashimo, starting from the shimo adjacent to the one from which the kete were taken. If the last kete falls into an empty shimo, the player's turn is finished. If the kete falls into an occupied shimo, the player picks up all the kete therein (including the last one played) and moves in the same direction as before. The player continues to put one kete into the adjacent shimo until the last kete falls into an empty shimo.

Example 1
As an example, let the state of one player's two rows be

1	3	0	2	1	4	0	2
2	0	1	1	1	2	2	6

If they select the shimo with 3 pieces and move clockwise then after the first sowing, the position becomes

1	0	1	3	**2**	4	0	2
2	0	1	1	1	2	2	6

In this matrix, the bold digit indicates in which shimo the last kete was dropped. We pick up the two kete and continue sowing to obtain this state:

1	0	1	3	0	5	**1**	2
2	0	1	1	1	2	2	6

At this point the kutakata move ends.

Example 2
In this example Mackay demonstrates the meaning of moving clockwise round the board:

1	3	0	2	1	4	0	**2**
2	0	1	1	1	2	2	6

starting from the marked shimo containing 2 kete, we obtain

1	3	0	2	1	4	0	0
2	0	1	1	1	2	**3**	7

then

1	3	0	2	1	4	0	0
2	0	1	**2**	2	3	0	7

then the move ends in the following state:

1	3	0	2	1	4	0	0
2	1	2	0	2	3	0	7

7.2 Trivial Never Ending Moves

As noted in [18] after playing kutakata a few times, you will find that it can take quite a long time for a kutakata move to end. In the last matrix above, which shimo when selected for kutakata, gives the longest move? One would be led to think from just looking at the board that the shimo with the highest number of kete is the one. That is obviously not the right answer as that shimo does not always determine that the last kete drops in an empty shimo!

This prompts the question "can kutakata go on for ever"? If so, what characterizes an initial state which can go on for ever? Or alternatively, can we prove that all kutakata moves come to end, whatever the initial state?

From here on we will deliberately stop hacking the initial arguments from David Mackay [2] and leave them intact. When we hack at it, we will let you know. David considers the above question not yet precisely posed since we have not mentioned how many kete may be on one player's two rows. We will leave this issue hanging, noting that it could be that there are self-perpetuating board states which involve more kete than are used in a normal game of Bao; so perhaps the rules of Bao have been chosen such that infinite loops are impossible. Often in theoretical physics, we find that a system changes from one sort of behaviour to another sort of behaviour as we increase the concentration of particles (for example, as we compress a gas, it turns into a liquid then into a solid); so one exciting idea is that perhaps Bao has a similar `phase transition' - maybe we'll find that infinite loops only happen when the number of kete is bigger than 137, or something like that?

From here on, Mackay leads you through a partial solution to the above question. His aim is to try to illustrate how theoretical research proceeds. There are many strategies that are helpful.

7.2.1 Strategy 1: Random exploration

It is essential to get a feeling for the problem. Try a few random board states and explore what it is that causes a kutakata move to terminate. Try to find initial states that don't terminate. Try to identify anything which increases or decreases steadily as kutakata proceeds -- if, for example, the number of empty mashimo increased steadily each sowing, then this might help us prove that every kutakata move must come to an end. I will assume you have explored this strategy. But don't spend too long on it.

Until we have resolved the basic question of whether there exist non-terminating states, it is important to try to attack the problem from both sides: try to find non-terminating states *and* try to prove that no such states exist.

It is good to think about what we are hunting for.

If a kutakata move goes on for ever, what happens to the board? Think about it. If a state is non-terminating, it must lead to a *repeating state* - that is, during kutakata, we must end up in a cycle of states that repeats periodically. This is true because there is only a finite number of states of the Bao board and after any sowing, the current state leads deterministically to the next state; because there is a finite number of states, all that can happen is we either reach an end-state or else we must eventually return to a state we have been in before! (here the `state' is a specification of the numbers of kete in all the mashimo, and a statement of which shimo is the one we are about to distribute, and in which direction). After returning to a state we have been in before, we must thereafter repeat the sequence of states exactly for ever and ever. This is known as a `periodic cycle'. Thus the search for non-terminating states is a search for self-perpetuating states - states which reproduce themselves after some number of drops.

7.2.2 Strategy 2: look at special cases

Look at special cases that are easy to think about. For example, board positions with very small numbers of pieces on the board. Put just two kete on the board; three; four; eight.

7.2.3 Strategy 3

We look at modified versions of the problem. Sixteen mashimo is a large number to think about. Let's work on a simpler version of the problem. Consider a smaller Bao board where each player has not sixteen mashimo but rather a smaller number M, say M=2, 3, 4 or 5, arranged in a circle as the 16 mashimo of Bao are. Can we find self-perpetuating states for these mini-Bao games?

As a general rule, don't be afraid of finding the solution to a different problem. It's always good to solve any problem, especially if it is an easier version of the original problem!

Consider the case M=2. Can you find a self-perpetuating state?
We are considering Bao with M=2 mashimo. An example of a board state that is not self-perpetuating is

$$2 \quad 2$$

because after one sowing (starting from shimo number 2) we obtain

$$3 \quad 1'$$

at which point the move ends. This board game state was implemented by Henney and Agbinya [3].

You can try to find self-perpetuating states by random search or by mathematics; or you can try to prove they do not exist mathematically. I recommend that you try both random search and mathematics. Here, we use mathematics.

7.2.4 Proceeding mathematically

The number of kete in each shimo is either even or odd. If the shimo we select contains an even number, then what is going to happen? Clearly our move will end, as in the "2 2" example above. Therefore to get a self-perpetuating state we need the selected shimo to contain an odd number. So the initial state must be of the form:

$$(a)(\mathbf{2m+1})'$$

where a and m are integers and **'** is used to indicate the state we end at.

If the state is self-perpetuating, then after one sowing, or some number of sowings, we want to end up in the same state, or else the state

$$(\mathbf{2m+1})'(a)$$

So, what is the state after one sowing, if we start from

$$(a)(\mathbf{2m+1})' \,?$$

It is

$$(\mathbf{a+m+1})'\,(m)$$

isn't it? So can you use the self-perpetuation condition to find a solution for *a* and *m* which is a self-perpetuating state, or is there no such solution?

Solution for M=2:
The state $3' \quad 1$ is self-perpetuating. After we have dropped the 3 kete, the first one in shimo 2, one in shimo 1, and one in shimo 2, we have $1 \quad 3'$. Clearly the next sowing brings us back to our initial condition.

Can you find a similar self-perpetuating state for M=3?

Hint: look back at the hint for the case M=2; there, we used a mathematical method to find a self-perpetuating state. Use the same method to write down an initial state

$$(a) \quad (b) \quad (c)'$$

and try to find one which, when moved, gives a state

$$(c)' \quad (a) \quad (b)$$

or

$$(b) \quad (c)' \quad (a)$$

Clearly, we need c either to be (3m+1) or (3m+2). Try both cases.

Solution for M=3

The state

$$4' \quad 2 \quad 1$$

is self-perpetuating or result to a never ending move.

7.2.5 Strategy 4: Generalization

What about M=4? What about M=16? Can you use your insights from M=2 and M=3 to find a self-perpetuating solution for *any* M?

Solution for M=4

The state

$$5' \quad 3 \quad 2 \quad 1$$

is self-perpetuating. Can you use your insights from M=2 and M=3 to find a self-perpetuating solution for *any* M? - In particular, M=16?

Solution for M=16

The state

17′	15	14	13	12	11	10	9
1	2	3	4	5	6	7	8

is self-perpetuating. After seventeen drops, the state looks identical to the starting state except it is shifted round by one place.

Note however this involves 137 kete and violates the rule that kutakata can only start from a shimo with 15 or fewer kete. We have thus come close to a solution. It looks as if Bao can go on for ever. **Now** the research problem is to establish whether all self-perpetuating solutions involve more kete than are actually used in a real game of Bao; or can we find self-perpetuating solutions that use fewer kete?

7.2.6 Strategy 4 (again): Generalization

Is there another way we could have generalized our solutions for M=2 and M=3 to get solutions for other values of M? Here again are the solutions we have made thus far:

$$\begin{array}{ll} M=2: & 3' \quad 1 \\ M=3: & 4' \quad 2 \quad 1 \\ M=4: & 5' \quad 3 \quad 2 \quad 1 \end{array}$$

The leap that is needed here may take you some time to think of; don't look at the hint unless you are really stuck.

Is there another way we could have generalized our solutions for M=2 and M=3 to get solutions for other values of M?

Hint: notice the way that the solutions for M=2, M=3, M=4 work; they involve a triangular shaped dart which crawls in the direction of motion of the kutakata, with the final kete always falling just on the large backside of the dart. Our solutions have involved just *one* dart going round and round. Could there be solutions with more darts following each other around?

Another way of making self-perpetuating states:
One idea is to have cascades of these darts: Here is a solution for the case M=11 that involves three darts of length 4 pursuing each other round and round the mashimo.

3 2 1 0 3 2 1 4′ 2 1 0

After one sowing this becomes

4′ 2 1 0 3 2 1 0 3 2 1

After twelve drops, the state looks identical to the starting state except it is shifted round by one place.

Congratulations!

We have almost solved the original problem. All we need to do now is come up with a version of the above solution for M=16.

First, can you come up with a solution for M=4 (different from the one we already considered)? In the case M=11, we squashed four darts of length three, with spaces (empty mashimo) between two pairs of them

3 2 1 0 3 2 1 4′ 2 1 0

and at any sowing, there is one extra kete in the active shimo. So how can we handle the case of M=4? We can put in darts of length one

1 0

or darts of length two

2 1 0

Noting that with three darts of length 3, we filled M=11 mashimo, how can we fill M=4? We can put a dart of length 1 alongside a dart of length 2:

1 0 2 1

then add one kete to the active shimo:

2′ 0 2 1

then when we takata,

0 1 3′ 1
1 2′ 0 2
1 0 1 3′
2 1 2′ 0

we have a periodic solution.

After five drops, the state looks identical to the starting state except it is shifted round by one place.

Congratulations! You are almost there!
Can you come up with a version of the above solution for M=16?
We can put darts of all sorts of lengths in the M=16 mashimo, so there are many solutions. Here are some:

A solution using mainly length 2 darts.

1	0	2	1	0	1	3′	1
2	0	1	2	0	1	2	0

A solution with some bigger ones:

4′	2	1	0	1	0	5	4
1	0	1	2	0	1	2	3

the sparsest such solution, that is, the one using fewest kete, is

0	1	0	1	0	1	0	1
2′	1	2	0	1	0	1	0

Congratulations!
We have solved the original problem. Bao, even with only 64 kete, can be put into a state in which kutakata lasts for ever.

7.3 Complex Never Ending Moves

The solution to one problem often makes us to pose new questions as to any other problems that are associated with the original problem or with the solution. Let us see if any more problems suggest themselves. MacKay posed the question whether we can create "self-perpetuating states that are not so trivial?" as in the above examples. All the above solutions repeat, with a shift of one place, every 17 drops. Can solutions be found in which the appearance of the state changes more radically? Can we identify solutions with periods larger than 17? If so what are those solutions?

7.3.1Solution

The following are some solutions found by MacKay which have periods bigger than 17. He found them by making minor alterations to earlier period-17 solutions.

0	2′	1	2	3	4	0	1
4	3	2	4	0	2	1	0

(the above has period 84)

3	4	0	2′	1	2	3	4
2	1	3	2	1	2	1	0

The following solution has an extremely large period. Can you work out what its period is?

0	1	2	3	2′	4	1	0
1	0	5	2	0	1	2	1

This is a very tricky question! The answer is 37179 drops or 12393 sowings (17×3^6 17).
What happens to the state

13	1	0	3	0	3	0	3
1	0	3	0	3	0	3	0

if you start kutakata at the 13? The answer to this is a periodic state with period 1836.

7.4 References

[1] Voogt, Alexander J. de, “Limits of the Mind: towards a characterisation of Bao mastership”, http://www.leidenuniv.nl/interfac/cnws/pub/alex.html, 1995.
[2] David Mackay, “Bao as a research problem”, mackay@mrao.cam.ac.uk, 1998
[3] Henney, Andre; Agbinya, Johnson; "African Board Games", Honours Project Report, Department of Computer Science, University of the Western Cape, Cape Town, 2003.
[4] Visa Korkiakoski and Janne Pänkälä, Warri (adaptation of mancala), http://www.hut.fi/~vkorkiak/mancala/doc/html_rules/

Chapter 8

OMWESO: Uganda's National Game

8 Introduction to OMWESO

Omweso is a board game mostly played in Uganda. It is also known as Mweso. It differs from the other board games in that the board consists of 32 holes distributed in four rows with equal number of holes. The Omweso board also does not have storage holes for players to store captured seeds. Each player has 32 playing seeds. The two players have a total of 64 seeds for the game.

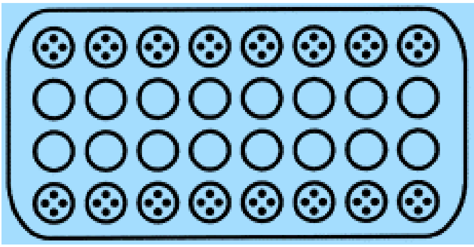


Figure 11: Omweso Board (Adapted from: Michael Sanderman)

8.1 Matrix View of Omweso Board

An empty matrix is used to represent the Omweso board below. Let $x_j(k)$ denote the number of seeds in hole k of row j (j=1, 2, 3 and 4) on the board game. Let the rows be numbered from the top as shown.

$$\begin{Bmatrix} & & A & B & C & D & E & F & G & H \\ P1 & row1 & . & . & . & . & . & . & . & . \\ P1 & row2 & . & . & . & . & . & . & . & . \\ P2 & row3 & . & . & . & . & . & . & . & . \\ P2 & row4 & . & . & . & . & . & . & . & . \\ & & I & J & K & L & M & N & O & P \end{Bmatrix}$$

The two players are denoted as P1 and P2. Player 1 (P1) uses row1 and row2 and Player 2 uses row3 and row4.

8.2 Start Up Stage

The start up stage of the game is not unique but varied. Players can set up in a manner that facilitates winning of the game. A typical start up stage is shown in this figure, whereby each player has set up the 32 sees in holes mainly in the front row of the board on their sides with an edge hole empty in the front row. This set up is typical of beginners to the game. This set up stage prevents an opponent delivering a knockout by "cutting off both his heads" or capture seeds from both ends of his board in one move at the beginning of the game.

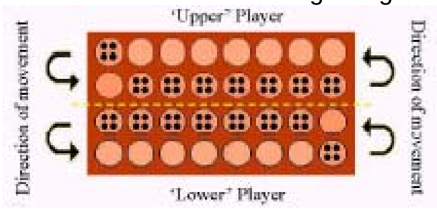


Figure 12: A novice's set up stage

This matrix shows another initial setup stage. The first row for each player has 4 seeds per cell. The second rows are initially empty, and represented below with dots. We could quite have easily put zeros in them and it will not alter the playing patterns or outcomes.

Figure 13:

$$\left\{\begin{array}{llcccccccc} & & A & B & C & D & E & F & G & H \\ P1 & row1 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ P1 & row2 & . & . & . & . & . & . & . & . \\ P2 & row2 & . & . & . & . & . & . & . & . \\ P2 & row1 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & & I & J & K & L & M & N & O & P \end{array}\right\}$$

This set up requires only two matrices F_1 and F_{1r}. Player P1 concatenates four of F_1 matrix and player P2 also concatenates four of F_{1r}. Therefore the set up process is:

$$P1 = F_1 \vdots F_1 \vdots F_1 \vdots F_1$$

and

$$P2 = F_{r1} \vdots F_{r1} \vdots F_{1r} \vdots F_{1r} \ .$$

8.2.1 Standard setups

Wernham presented examples of standard set up in the game. He gathered them through informants. Two groupings are described.

$$\left\{\begin{array}{llcccccccc} & & A & B & C & D & E & F & G & H \\ P1 & row1 & & & & & & & & \\ P1 & row2 & 7 & 3 & 3 & 3 & 3 & 3 & 3 & 7 \\ P2 & row2 & 5 & & 7 & 2 & 4 & 2 & 6 & \\ P2 & row1 & & & & & & & & 6 \\ & & I & J & K & L & M & N & O & P \end{array}\right\}$$

Junior grouping is preferred for new and relatively young players. The groupings are based on the number of seeds deployed per hole. In junior grouping, there are fewer seeds in each hole with a maximum of 5, 6 or 7 seeds usually. In this figure, P2 has countered the set up by P1. When P1 starts play by sowing seeds into the back row, P2 is able to counter by also sowing and depositing seeds into own back row as well.

Exercise: Write the start up matrix depicted above in terms of defining matrices

Senior grouping uses fewer holes at start with more seeds in a particular hole to enable sowing round the board on own side. It is typical to see 23, 22, 21, 20, 19 and 17 in one hole with the rest of the seeds deployed in other neighbouring holes. Here is an example:

$$\left\{ \begin{array}{lcccccccc} P1 & 23 & & & & & & & \\ P1 & & 3 & 3 & 3 & & & & \end{array} \right\}$$

To counter this set up P2 has several options left to him and a few examples are given below.

$$\left\{ \begin{array}{lcccccccc} P2 & & & & & 3 & 3 & 3 & \\ P2 & & & & & & & & 23 \end{array} \right\}$$

or

$$\left\{ \begin{array}{lcccccccc} P2 & & & & 3 & 3 & 3 & 3 & \\ P2 & & & & & & & & 20 \end{array} \right\}$$

or

$$\left\{ \begin{array}{lcccccccc} P2 & 1 & 1 & 2 & 3 & 3 & 18 & & 2 \\ P2 & & & & & & & & 2 \end{array} \right\}$$

Here we observe a frontal defense and an equally rear back up by P2 to the setup by P1. This allows P2 to respond to the sowing of seeds by P1 appropriately.

Exercise: Transform the above matrices in terms of game defining matrices.

8.3 Rules of the Game

We will now list the rules that govern playing Omweso. Omweso permits captured seeds to be re-sown into the board. Player plays in an anti-clockwise movement. In tournaments, there might be referees. Since in such situations, the rules may be agreed by tournament organisers in an ad hoc manner and may vary from region to region and tournament to tournaments, we have not covered such rules in this book.

8.3.1 Sowing Seeds - Rule 1

Choose a hole from your side of the board to use in starting the game. Then pick all the seeds in that hole and sow them in singletons in an anti-clockwise manner in subsequent holes. If the last seed falls in an empty hole (one seed or singleton in the hole), your turn ends. Singletons cannot move. If it does not, scoop the seeds in the last hole and continue to sow until one of your seeds falls in an empty hole.

For instance if player 2 is elected to play first and decides to sow from cell K, then he/she will sow from left to right in row 1 of his/her rows.

Figure 14: Beginning of Play by Player P2

$$\left\{\begin{array}{cccccccccc} & & A & B & C & D & E & F & G & H \\ P1 & row1 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ P1 & row2 & . & . & . & . & . & . & . & . \\ P2 & row2 & . & . & . & . & . & . & . & . \\ P2 & row1 & 4 & 4 & 0 & 5 & 5 & 5 & 5 & 4 \\ & & I & J & K & L & M & N & O & P \end{array}\right\}$$

If player 1 selects to play to sow from cell B, then he/she will sow from left to right in row 1 of his/her rows.

8.3.2 Relay Sowing - Rule 2

In Figure 10, the last seed did not fall in an empty hole. Therefore the player continues play until he finally ends by sowing the last seed in an empty cell (Figure 11) or a capture is possible.

Figure 15: End of Play for Player P2

$$\left\{\begin{array}{cccccccccc} & & A & B & C & D & E & F & G & H \\ P1 & row1 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ P1 & row2 & . & . & . & . & . & . & . & . \\ P2 & row2 & . & . & . & . & 1 & 1 & 1 & 1 \\ P2 & row1 & 4 & 4 & 0 & 5 & 5 & 5 & 0 & 5 \\ & & I & J & K & L & M & N & O & P \end{array}\right\}$$

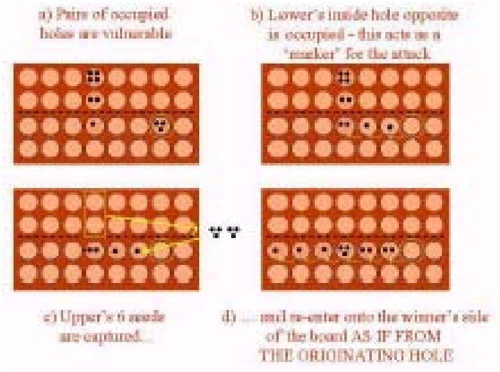
This process of continuity of sowing of seeds without the other player having a turn is called relay sowing. The rule is, if the last seed is not sown into an empty hole, pick up the seeds in the last hole sown and re-sow them into the subsequent holes, i.e. sow again. As long as the last seed does not fall into an empty hole, sow again and again and again until an empty hole is found.

8.3.3 Capture - Rule 3

If you have a pair of holes with seeds in them, they are vulnerable. A player is compelled to make a capture if it is a resultant move. It is however not necessary for a player to make a move that will result in a capture. A capture is only possible if the following conditions are met (Figure 12).

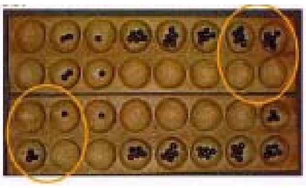
- ❖ If the last seed is sown in a cell, and that cell contains seeds.
- ❖ This particular cell can only be in row 2.
- ❖ The opponent's pair of cells directly opposite the player's cell contains at least 1 seed each.

Figure 16: Capture Situations

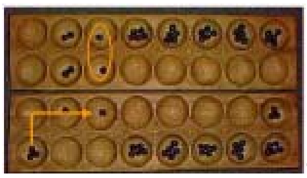


8.3.4 Reverse Capture - Rule 4

Reverse capture is allowed from the left most four holes and clockwise movement from these four holes is also allowed if and only if capture is possible. A reverse capture situation is shown in Figure 13.



Reverse capture



Clockwise move

Figure 17: Reverse Capture States

8.3.5 Winning

Winning is the objective of most games. It does not matter what people say, they are there to win, not just to have fun. When someone loses a game they say they were playing for fun only, what happens? Tempers will flare if you joke about their losing! There are several ways of wining a game of Omweso.

Akakyala - capturing seeds from the loser in two separate moves before the loser has even made the first capture of the game. This is precisely one reason why setting up properly at the start of a game of Omweso is very important. Setting up badly provides the grounds to lose the game. The reverse is also very true.

Emitwe-Ebiri - "cutting off at 2 heads". In this case the winner captures both extreme pairs of holes in one move.

When, the loser is immobilised and unable to move. The loser either has empty holes or singleton seeds. These acts of loses are very typical of warfare situations, an empty village, a lone survivor remaining, or an army whose leadership is defeated from the front, the rear and the flanks, or being caught napping at home before an army is ready to mobilise to fight. These acts of warfare are symbolised in the game of Omweso.

8.4 References

[1] Michael Sanderstand

Chapter 9

The Rules of Bao

9 Introduction

Bao ("wood" in Kiswahili) is played on a board having 32 holes and 64 beans. The seeds are arranged in the 32 holes in four rows of eight. Two rows belong to one player, and two to the other. The holes are called mashimo (holes, singular, shimo) and the beans are called kete (singular kete). Here is a view of the board, with each "O" being one shimo

	P	***l***	***a***	***y***	***e***	***r***		***A***
Back row of A	0	0	0	0	0	0	0	0
Front row of A	0	0	0	0	0	0	0	0
Front row of B	0	0	0	0	0	0	0	0
Back row of B	0	0	0	0	0	0	0	0
	P	***l***	***a***	***y***	***e***	***r***		***B***

The Bao rules reproduced in this section are found in [1]. The game reproduced from [19] is that variation of "Bao" which is played on the coast of East Africa. The kete in the two rows of mashimo nearest to each player belong to him. Each player in turn moves his kete with a view to capturing his opponent's kete and transferring them to his own mashimo. It is necessary to demonstrate the basic moves. For this purpose the learner should distribute kete indiscriminately over the board.

9.1 KUTAKATA

1. This move is made when a player is unable to capture any kete of his opponent. Select any one of your mashimo and pick up all the kete from it: moving clockwise or anti-clockwise around your two rows, put one kete in the adjacent shimo and one in each of the following mashimo until none remain in the hand. If the last kete falls into an empty shimo, the turn is finished: if it falls into an occupied shimo, pick up all the kete therein (including the last one played) and moving in the same direction as before put one kete in the adjacent shimo and one in each of the following. Continue in this way until the last kete in hand falls into an empty shimo.

2. The following points should be noted:

   (a) You may not start to takata with only one kete.
   (b) You may not start to takata with kete from a back-row shimo if you can do so with kete from a front-row shimo.
   (c) Having started to takata in one direction you may not change direction in that turn.
   (d) Having started to takata you cannot capture any kete during that turn.
   (e) You may not takata if you could capture any kete of your opponent.

9.2 CAPTURING

We shall use the word "marker" to describe an occupied shimo of the front-row which is in line with an occupied shimo of the opponent's front-row.

3. "Mtaji" (plural "mitaji") is the term applied collectively to the kete (not less than two nor more than fifteen) contained in one shimo, the last of which when distributed one by one in either direction will fall into a marker.

4. To capture, you select a mtaji and commence your move as in kutakata. However instead of picking up your own kete from your marker, you pick up your opponent's kete from his marker. The captured kete are then distributed in the following manner.

5. If the opposing marker was in one of the four central lines, and you were moving clockwise, put the first captured kete into your left-end front-row shimo, the next kete into the adjacent shimo on the right, and so on in a clockwise direction around your two rows. If the opposing marker was in one of the four central lines and you were moving anti-clockwise, put the fist captured kete into your right-end front-row shimo, the next kete into the adjacent shimo on the left, and so on in an anticlockwise direction.

6. If the opposing marker was in either pair of the four outer lines, put the first captured kete in your nearest end-shimo to that marker, the nest in the adjacent front-row shimo and so on, clockwise or anti-clockwise as the case may be.

7. If the last captured kete falls in an empty shimo the turn is finished. If it falls into a marker, capture and proceed as in paragraphs 5 or 6.

8. If the last captured kete falls into a shimo which is neither empty nor a marker, proceed as in paragraph 1. This must not, however, be referred to as "kutakata" because you can again capture during the same turn whenever the last kete in hand falls into a marker.

9.3 BAO LA KUJIFUNZA

9. Having learnt these moves the learner should now place two kete in each shimo and play a simplified form of bao with an opponent. The first player selects a mtaji and captures two kete; if he has selected badly his turn is now finished, but if he has selected well he will capture a further six kete and leave his opponent only five mitaji to choose from.

10. The game continues until the losing player either has no kete in his front-row, or has no mtaji and nothing to takata with.

11. The skill consists in seeing a number of moves ahead, and in attacking the opponent's mitaji by capturing either the mitaji themselves or the kete of the appropriate markers. When forced to takata for lack of an mtaji it is usually best to select the shimo with the largest number of kete.

9.3.1 BAO LA KISWAHILI

12. In the East African Coast fame of bao only twenty kete are placed on the board at the start of the game, the remaining kete (the stock) being brought into play one at each of the fist forty-four turns. When all the kete are in play the game continues and finishes as in "Bao la Kujifunza" above.

13. To set out the board each player places six kete in his front-row shimo immediately to the right of the centre of the board, and two kete in each of the next two mashimo to the right (leaving the end shimo empty). The six kete, together with any further kete added to them during the play, are called collectively the "nyumba" of each player. In some boards the mashimo which initially contain the nyumba are enlarged.

14. The first player takes a kete from the stock and adds it to one of his two pairs of kete on the board; he may add these three kete to the two kete on the right or left as the case

may be, and takata in the same direction with all five kete the first of the five being placed either in the nyumba or in the right-end shimo.

15. If he has a marker, the second player must put a kete from the stock into his marker thereby capturing one of his opponent's kete which he places in either of his end-mashimo. If he has more tan one marker he chooses his nyumba for obvious reasons. If he has no marker he plays as in paragraph 14.

16. Each player in turn continues to add a kete from the stock to one of his occupied mashimo of the front row in accordance with the following rules; which apply to the first part of the game

    (a) If he has a marker he must capture and proceed as in paragraphs 5-8
    (b) If he cannot capture he must takata
    (c) While any kete remain in the stock, a player may not commerce to takata with the kete of a back-row shimo, though he may continue tot takata with them
    (d) He may neither commence nor continue nor continue to takata with his nyumba
    (e) if he has no marker and cannot takata, he must put his kete from the stock into his nyumba, take two kete from the nyumba place one in each of the next two mashimo to the right or the next two mashimo to the left
    (f) If having captured during that turn, the last kete in hand falls into the nyumba, the player may at that point stop his turn or he may distribute his nyumba in accordance with paragraph 8
    (g) The first part of the game proceeds until all the stock has been brought into play or until the losing player has no marker, cannot takata, and has no nyumba left. This will be either because he has distributed his nyumba or because it has disappeared bye reason of paragraph 16 (c)

17. If the game has not been brought to a close during the fist part, the rules applicable to the second half are those described in paragraphs 1-11.

9.4 Reference

[1] Victor Bautista I Roca, “The rules of bao”, http://drac.com/pers/viktor/bao.html, 1996.

Chapter 10

Minimax Algorithms in African Board Games

10 Introduction

Game playing is mostly a search for the best moves to result to a win. There is an inherent dilemma in every game. Most of the time when one makes a move, the opponent is most likely to also make a move, but none of the two players is definitely sure of what the state of the board will be after those moves. This area of study is well known in computer science as artificial intelligence and has been studied by many researchers. Since your opponent will always try to thwart every move you take, what is the best approach to use in reducing the effects of such attempts? For any counter moves to be effective there has to be a means for evaluating or measuring how bad or good your present game state or position is.

The action of an opponent in games like Bao or Omweso can be estimated using *minimax* algorithm. A utility function is then used to measure the outcomes of moves in the game. The *minimax* algorithm is based on using a search tree to find the best next move in a given turn of play and on a given set of possible moves. One of the main focus of each game is deciding the best move. This is often preceded by players counting seeds in their hands if they are allowed to re-sow them and/or counting holes on the board to determine the extent to which a set of seeds in a hole can reach.

To decide which move is optimal the computer player uses an algorithm to find out what the next best move of the human player would be in the situation at hand.

To be of real advantage to the computer player, the strength of this best move by the human player should be as low as possible. This best move of the human player is the one that results in the smallest possible number of good moves for the computer player. The more the computer can remember what would happen in future as a result of a move, the better the outcome of the algorithm to the computer. In practice the computer's memory is of limited size, so at a certain depth in this search tree you just use the so-called *static evaluation function (SAF).*

The strength of some the moves can be judged by how many recursive sowing it permits or how many seeds from the opponent's side can be captured. Alternatively in situations where defeat stares the player at the face, the strength of a best move could be that which leads to a stalemate. The algorithm should therefore minimise disadvantages to the player and maximise his gains from the move. This is why the algorithm is called the minimax algorithm. It is concerned with finding the move towards the situation that has the minimum or maximum value, depending on which side has to make the move.

The application of minimax algorithm is very prominent in African board games. Certain game defining matrices are preferred to start games of Omweso and Bao simply because they provide the player a minimax gain. The optimum defining matrices look ahead into several steps in the game from start up to winning a game. Most of the expert players are those who are capable of innate ability to look ahead through the board and the seeds sown or to be sown and determine their best moves and starting points.

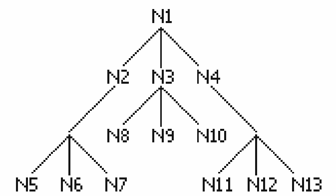
Anyone new to the Artificial Intelligence field with an interest in gaming should attempt to create a simple board game program with an AI opponent. The best part about board games are the simple rules - this means less time implementing the game, more time on the AI. This essay will detail some of the techniques that you can apply to *simple* board games (this does not include chess, Go, Go-moku and the likes).

10.1 Minimax Trees

The discussion in this section is based on the paper by James Matthews [xx]. We first discuss how to maximize our gains from each move and minimize the benefits to our opponents.

10.1.1 Max

In this example we imagine a simple game and at each node (game state space) there are only three moves, one move results in a win, another a draw, and finally a loss. Therefore, we want to assess each node and figure what the outcome will be. Let us extend our game to one that allows you to make three moves per go, and only takes two layers (goes) to win. Therefore, we will want to look ahead to find out which move combination works for us. We can generate a *game tree* of all the possible moves as shown in this figure.



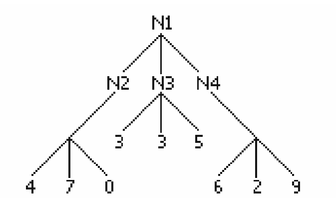
You can see how there are 9 final possible moves. Imagine that N11 is the winning situation, therefore our first move will have to be N4. How can we figure this out algorithmically? Let us assign values to a win, draw and loss. A win will be 1, a draw 0 and a loss -1. Say that N11 is the only winner, and the rest are drawing situations. So, what we will want to do is evaluate the tree from the bottom-up propagating the *maximum* value for the nodes upwards. Therefore, for the N5-N7 group, 0 is the highest so this is applied to N2. N8-N10 also has 0 as the highest, which is taken on by N3. The N11-N13 group has 1 as the highest. The program knows to choose N4.

In our example, since the tree is only two layers deep this seems rather trivial. But imagine a tree 10 layers deep, this method would allow you to simply calculate which moves would lead to a winning situation. Most of you will already notice a large fault in this - trees this large are incredibly expensive in both memory and computational terms. A 10-layer tree that branches three times for each node would have 59,049 nodes. This is relatively simple.

Therefore, we have to cut down the depth of our tree. This gives us a problem, though - if we limit the depth we are not guaranteed a winning scenario as one of the nodes. This is where clever programming has to come in. You must create some sort of evaluation function that can assess how close to a winning situation the board is. Since the nodes are not so clear-cut (win, draw, loss) a more complicated numbering system has to be used. The system is completely dependent on the programmer and the board game in question. This is only half the problem, though, because we have not yet factored in an opponent player.

10.1.2 Min and Max

Very few board games are one player, so how can we add this into our tree? Think about it, when we are playing for ourselves we are attempting to *maximize* our score, so our opponent will want to *minimize* our score. Let us look at a game tree - this one though is limited to a depth of 2. Here is our game tree with evaluations assigned to the final nodes:



Now, the assigned values are for boards representing our opponent's choice of boards. N1 stands for the current board, N2-N4 are our three possible moves and N5-N13 are the opponents possible follow-up (reply) moves. Our opponent will try to minimize our winning possibility. Therefore he will calculate the *minimum* value for each node and assign it to the parent:

- N2 will equal 0
- N3 will equal 3
- N4 will equal 2

Of course these values are the smallest values attached to the leaves of the tree at the layer being considered. Finally, in making a choice for our best possible move, we look at the maximum of these values - which is 3 (N3).

10.1.3 Conclusion

In conclusion, let us consider the steps involved in the algorithm:

1. Generate the game tree to depth *d*.
2. The final nodes (leaves) should have values assigned to them, denoting how close to a winning scenario they are. The higher the value, the closer to a winning scenario.
3. Then minimize the scores if calculating for the opponent, and maximize them for the player
4. Propagate the scores upward through the tree until the computer can make a decision as to the best "branch" of game play to explore.

The minimax algorithm is surprisingly simple, but its downfall lies in the huge amount of memory and computation it can take to achieve decent results. There is a lot of research into minimizing the expansion of the game tree. The most popular pruning method is *alpha-beta pruning*. The bottom line is, try to minimize the number of branches for each node and keep the depth at a reasonable level. Remember the number of nodes required is equal to b^d , where *b* is the number of branches per node and *d* is the depth of the tree.

10.2 Board Game Design: Immediate Moves

In this section we look at immediate moves rather than look into the future. Most moves are immediate with no thoughts for the outcomes or repercussions from the move. Most human non expert players use a lot of on the spot moves and have no real strategies in place to deal with the consequences. Computers however think ahead and about the consequences of moves. These are in built into the software by the software designers.

How does the computer deal with the current situations in deciding the best move and to make a move? We can answer this question by using a simple example. This question was answered by James Matthews using the following example of a priority board.

A priority board is a board identically sized board to your game board that the artificial intelligence agent uses to store values in. Each value corresponds to the place on the board. The higher the value, the more likely the agent will move there. A priority board can be very useful to assess the large number of places in games like Bao and Omweso. So, how do you assess the positions?

10.3 Heuristics

Heuristics is basically a fanciful name for rules and heuristics make or break a game. Firstly, you should narrow down your board game rules to a couple of heuristics that you can implement as code.

For example, the rules of Pente dictate that you win if you place 5 pieces in a row, or get 5 captures. Therefore, some possible heuristics are as follows:

- Check for 4-pieces in a row
- Check for 3-pieces in a row
- Check for 2-pieces in a row
- Check for captures

Now, obviously, the first 3 can be reduced further, "Check for x-pieces in a row" which narrows our heuristic down to two! Thing is, so far we have only an offensive agent. In the worst case scenario, we'd want a purely defensive agent - *purely offensive agents will be easily beaten!* So, we will add an additional 2 heuristics to our program:

- Check for opponent potentially getting x-pieces in a row
- Check for opponent potentially making a capture

We now have 4 heuristics that can be applied to the game in both defensive and offensive styles. I have found the best place to proceed from here is by applying a two step evaluation. Firstly, you assign a number to the priority board place in your heuristic routine. For example, in our "Check x-pieces in a row" heuristic we could assign the number of pieces to the priority board. For example, given a simple Pente board:

```
[ ][ ][ ][ ][x]
[ ][ ][ ][x][ ]
[ ][ ][ ][ ][ ]
[ ][x][ ][ ][ ]
[ ][ ][ ][ ][ ]
```

Our initial priority board after the first heuristic would look something like this:

```
[1][2][2][2][0]
[1][1][1][0][2]
[1][1][3][1][2]
[1][0][1][2][1]
[3][1][1][1][1]
```

Note that the diagonal has the highest values since there are 3 pieces in the 5-line, therefore (3,3) and (5,1) get the highest values assigned to them. With experience, I've found it useful to multiply each heuristic by a certain *bias*. I've also found it useful to assign a *slightly* higher value to defensive heuristics (for example, I use a bias of 1.2 as opposed to 1.0 (which is no bias) in PenteAI).

Hopefully, you will find with the right heuristics and biases, you will get a certain amount of emergent behaviour - behaviour you had not planned, but comes from all the heuristics interacting with each other. For example, when I first ran my recent PenteAI program, I found the AI spaced out its moves. I had not planned this at all, indeed I wanted quite the opposite! After playing with this for a while, I realized it was an excellent way of limiting your opponents moves 3 or 4 moves ahead - and it is a technique that I myself have now adopted when playing the game!

10.4 Beyond This...

For board games this simple, there aren't that many additional simple techniques. It all depends on your game and the complexity of the game itself. For more complex games, you start to look a certain number of moves ahead to keep in front of your opponent. You can also employ more advanced AI techniques to simple board games with some interesting results - genetic algorithms to evolve biases and weights (although this is time consuming) can generate a very formidable opponent. Using neural networks to recognize patterns in the players behaviour can also be very challenging for the player.

10.5 Reference

[1] James Mathews, “Simple Board Game AI”

Appendix I

Game Links

- Awale (http://www.xkee.com/game/awale/)
- Awari (http://awari.cs.vu.nl)
- Dara (http://www.rekenwonder.com/dara.htm)
- Davis, J.E., and G. Kendall. Preprint. An investigation, using co-evolution, to evolve an awari player. Available at http://www.cs.nott.ac.uk/~gxk/papers/cec2002gxk.pdf.
- The complete rules of awari can be found at http://www.awari.com/ or http://www.cs.ualberta.ca/~awari/rules.html.
- Play awari online at http://awari.cs.vu.nl/.
- See http://www.myriad-online.com/awalink.htm for links to additional Web sites where you can play the game or download software for playing it on your computer or even cell phone.
- The University of Alberta computer awari group Web page at http://www.cs.ualberta.ca/~awari/.

Chapter 11

NeuroWarri: Using Evolutionary Computation to Develop Warri Strategies [9]

11 NeuroWarri Abstract

Warri is an ancient count-and-capture game that is played throughout the Caribbean, especially in the islands of Barbados and Antigua. Several computer-based versions of this game are now available, in both free and commercial forms. These games are generally based on common methods of computer game playing such as minmax trees and alpha-beta search. They also use various heuristics methods that have been created specifically for Warri. As a result, research has been focused on developing stronger evaluation functions for these games. For example, many games strengthen their evaluation functions by using move-databases, which contain moves for the start, middle and end game. This paper discusses the development of a neural network computer-based version of the Warri game, called *NeuroWarri*. Using evolutionary computation, the system was able to learn the rules of the game and develop its own strategies without human intervention.

Keywords: Evolutionary computation, neural networks, Warri.

11.1 Introduction

The Barbados version of Warri is one of the oldest surviving games in the island. It originated in the Kush Civilization of the Upper Nile and is one of many count-and-capture games of that time. During the slave trade in the 17th century, the game was brought to the Caribbean [6,7]. The Barbados version is based on the Asante version, Oware, where the rules have been carefully preserved over the years.

Warri is based on moving seeds around a board, capturing your opponent's seeds until you have gathered enough to declare a win. In spite of its simplicity, it tests both your mathematical skills and strategic planning ability. There are in fact several variations of these count-and-capture games throughout the world but the majority can be found in Africa, Asia and the Caribbean. However, all are based on a common theme of moving seeds around a board and vary in how the seeds can be captured and the winning conditions.

There are several free and commercial versions of these games available. The more popular and well-known ones can be found on the University of Alberta Awari web site [5]. In most cases, the more popular variations of Oware are provided, with Barbados Warri being one of them. There is also research currently being conducted at the University of Alberta in the area of game playing [5]. While the actual algorithms are generally kept hidden, those in the research arena seem to be based on standard methods of computer game playing such as minmax trees, alpha-beta search and various heuristics specific to the game. However, in all of these cases, the objective is to develop the strongest evaluation function for Warri.

With this background, NeuroWarri was developed. The primary objective was to explore the ability of a computer to learn the rules and strategies of Warri using evolutionary computation. NeuroWarri's evaluation function is based on a neural network that is trained using the evolutionary-computational methodology created by Fogel in the design of his draughts (checkers) game [2,3,4].

This paper presents the overall design and implementation of the NeuroWarri system. Section 2 describes the rules of Warri. The evolutionary process is described in sections 3 to 5 while the strategies developed by NeuroWarri are dealt with in section 6. Section 7 discusses the testing and the implementation of the game while section 8 contains suggestions for future enhancements.

11.2 The Rules of Warri

The version of Warri used by NeuroWarri is the one commonly played in the island of Barbados. This game consists of a board made up of twelve houses divided into two rows of six, as shown in figure 1. Each side belongs to one of two players. The game starts with four seeds in each of the twelve houses, giving a total of 48 seeds. The objective of the game is to capture the majority of the seeds with the game ending when a player has gained 25 or more seeds. A draw can also occur if both players capture 24 seeds. Some versions of Warri require the players to play until all of the seeds have been removed but in the case of the Barbados version, the game ends when a player gains 25 or more seeds. In the case of a draw of less than 24 seeds, the game ends after a specified number of moves or when the moves are repeated.

```
          X | 0
 f    e    d    c    b    a
------------------------
 4 | 4 | 4 | 4 | 4 | 4
 4 | 4 | 4 | 4 | 4 | 4
------------------------
 A    B    C    D    E    F
          Y | 0
```

Figure 1: The Warri Board

Consider figure 1, there are two players, X and Y. The two boxes above and below the board contain the seeds captured by either player, with X at the top and Y at the bottom. The top row of houses belongs to X and the bottom to Y. The houses are labeled from A to F, with A being at the left side of each player. In this paper, each move is denoted by the name of the player followed by the label of the house. For example, X.a means that X moved the seeds in it's 'a' house.

A player moves by taking the seeds from a house on his/her side and placing them into houses to the right of the selected house. One seed is added to each house encountered. This continues in a counter clockwise direction until all of the seeds have been used. Consequently, a player can place seeds into an opponent's house. If the user encounters the house from which the seeds were originally taken, the house is skipped. This ensures that the starting house remains empty for that particular play. Consider figure 1, if the move Y.E is made then the board is as follows (see figure 2).

```
          X | 0
 f    e    d    c    b    a
------------------------
 4 | 4 | 4 | 5 | 5 | 5
 4 | 4 | 4 | 4 | 0 | 5
------------------------
 A    B    C    D    E    F
          Y | 0
```

Figure 2: Move Y.E

A capture is made if the last seed placed on the board falls into a house that contains one or two seeds and is on the opponent's side. The player then takes all of the seeds in that house. Therefore, a player can capture two or three seeds per house. If the preceding house also contains two or three seeds then these are removed. This process continues in a clockwise motion, until either all of the opponent's houses have been dealt with or a house was encountered that did not contain two or three seeds. As a result, a house is vulnerable if it contains one or two seeds. Consider figures 3, 4 and 5.

```
        X | 10
 f   e   d   c   b   a
------------------------
 0 | 1 | 2 | 1 | 3 | 0
 0 | 5 | 0 | 0 | 6 | 0
------------------------
 A   B   C   D   E   F
        Y | 20
```

Figure 3: Before Capture (Before the Move Y.E)

```
        X | 10
 f   e   d   c   b   a
------------------------
 0 | 2 | 3 | 2 | 4 | 1
 0 | 5 | 0 | 0 | 0 | 1
------------------------
 A   B   C   D   E   F
        Y | 20
```

Figure 4: During Capture

```
        X | 10
 f   e   d   c   b   a
------------------------
 0 | 0 | 0 | 0 | 4 | 1
 0 | 5 | 0 | 0 | 0 | 1
------------------------
 A   B   C   D   E   F
        Y | 27
```

Figure 5: After Capture

In figure 3, the move Y.E is made and 6 seeds are distributed in a counter clockwise direction. In figure 4, the last seed lands in X.e and brings the total to 2 and so these are captured by Y. The preceding house, X.d, contains 3 and so these are also captured by Y. The same applies to X.c. However, in X.b, the total is greater than 3 and so the capturing ends. Note that Y now has a total of 27 seeds and so is declared the winner.

A penalty is imposed on a player if he/she makes a move that leaves the opponent with no seeds. If this happens, all of the remaining seeds on the board go to the opponent. Therefore, a user that makes this move must ensure that the total seeds gained by the opponent does not result in a score high enough to win. Consider figures 6, 7 and 8.

```
X | 20
 f   e   d   c   b   a
------------------------
 0 | 0 | 0 | 1 | 1 | 1
 0 | 5 | 0 | 0 | 0 | 3
```

```
 -----------------------
  A   B   C   D   E   F
          Y | 17
```

Figure 6: Before Penalty (Before the Move Y.F)

```
          X | 20
  f   e   d   c   b   a
-----------------------
  0 | 0 | 0 | 0 | 0 | 0
  0 | 5 | 0 | 0 | 0 | 0
-----------------------
  A   B   C   D   E   F
          Y | 23
```

Figure 7: During Penalty

```
          X | 25
  f   e   d   c   b   a
-----------------------
  0 | 0 | 0 | 0 | 0 | 0
  0 | 0 | 0 | 0 | 0 | 0
-----------------------
  A   B   C   D   E   F
          Y | 23
```

Figure 8: After Penalty

In figure 6, the move Y.F is made, resulting in Y capturing the seeds in X.a, X.b and X.c, giving a total of 23. However, X has no seeds to move and so will capture all of Y's seeds (see figure 7). This results in a total of 25 for X, making X the winner (see figure 8). A better move would have been Y.B instead of Y.F.

11.3 Developing an Evaluation Function

One of the main challenges with developing computer-based Warri games is that the game-designer needs to discover and list the fundamental principles used by the experts of those games. These principles are then encapsulated into an evaluation function that is used to determine which moves will produce the best results for the system. The system is then tested against human opponents and based on the results of actual game play, the evaluation function is further modified. This is repeated until the desired results are obtained. In many cases, the system is tested against other computer-based versions as in the world Warri games at the Mind Sports Olympiad in 1990, 1991 and 1992 [8].

The main disadvantage of this technique is that it requires the developer to have a considerable knowledge about the game. This is usually acquired by observing experts in the game and by playing against other Warri players. The information gathered includes the capturing of pieces, long term strategies, begin and end games, blocking techniques and attacking versus defensive moves. On discovering these techniques, converting them into a usable evaluation function provides another challenge. A variety of factors need to be considered, such as how the board will be encoded and the values given for piece positions.

It would of course be easier if this evaluation function could be discovered with as little human intervention and effort, as possible. Neural networks allow you to accomplish this. Traditionally, a neural network is provided with a set of input and output pattern pairs. Using this information, the network can be trained so that it will produce the correct output for a given input. Consequently, the neural network represents the desired evaluation function. The advantage of this method is that you do not need to give the network all of the possible patterns and, the resultant network will generally form a high quality evaluation function, with little human intervention.

However, using neural networks for computer-based games poses its' own problems. Firstly, it is extremely difficult and in some cases impossible, to create a large enough set of patterns that will produce a usable neural network. Also, the patterns will not take into account various strategies that could be employed during the game. In the case of Warri, there are only six possible moves a player can make, however, the number of possible patterns can grow rapidly since there are 48 seeds that can be spread among the twelve houses. Another factor is the strategies employed during the game. Despite the simplicity of Warri, a player must employ mathematical and logical skills, as well as the ability to plan forward thinking plays. For example, should the player build up a house or use an attacking strategy?

Evolutionary computation provides a solution to this dilemma. It allows the neural network to be evolved using the most basic information. The network can learn the rules of the game as well as develop its own strategies without human intervention.

11.4 Structure of the Neural Network

The evaluation function for NeuroWarri is represented by a backpropagation neural network with a single hidden layer and a single output (see figure 9) [1]. The output contains a bias of the total seeds captured by the player minus those of the opponent. This was done so that the neural network understood that the total seeds captured is important and that it should try to maintain this difference in its favour i.e. it should be a positive value. The only other information provided to the network was that moves could not be made with houses containing 0 seeds.

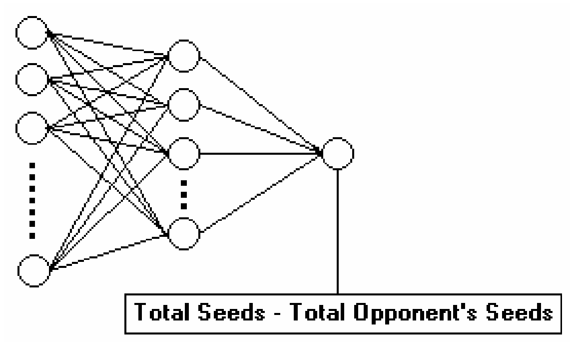


Figure 9: NeuroWarri Neural Network

There are 84 inputs used to represent the total seeds in each house and the total captured by the players. In order to encode this information for the neural network, the board's upper and lower sections was laid out end to end with the lower end occurring first (see figure 10). Each house was represented by six of the inputs. Using these inputs the total number of seeds in each house can be represented in a binary format. The six inputs gave a maximum value of $2^7 - 1$ or 127. However, the total pieces on the board is 48, and so this large number allowed for future expansion of the game. This gives a total of 72 inputs (6 * 12), leaving 12 unallocated inputs (84 – 72 = 12). The rest of the inputs were allocated as follows.

- The binary representation of captured seeds used 5 inputs.
- The binary representation of seeds captured by the opponent used 5 inputs.
- The last two inputs indicated which player was being considered, where one player is represented by the binary value 10 and the other 01.

A	B	C	D	E	F	f	e	d	c	b	a

Figure 10: Board Placed End to End

The hidden layer contained 40 nodes in total. This single hidden layer was found to provide an adequate level of performance.

11.5 Evolving the Neural Networks

The neural network was trained using evolutionary computation in the form of a competition. The system consisted of 20 neural networks whose weights were initially randomly generated. Each network represented a player in the competition. Each player played 5 games against 5 randomly selected opponents. A win obtained a positive score (greater than 0), a draw 0 and a loss, a negative score (less than 0). The absolute value chosen for the positive score was always less than that of the negative score. This ensured that that the emphasis was placed on the number of losses made so that only the strongest networks would be selected.

After the competition, the 10 players with the highest score were selected and used to create 10 new sibling networks. A sibling network was created by perturbing the parent network's weights using the algorithm developed by Fogel for the Checkers game [2]. The actual weight change equation used by NeuroWarri is given below.

$p' = p \times \exp(vN(0,1))$
$n = 1$
while $n \leq W_T$ **do**
  $\Delta w_n = p'N_n(0,1)$
  $w'_n = w_n + \Delta w_n$
  $n = n + 1$
end

Where,

- w_n is a weight in the neural network
- p' and p are temporary variables which contain the weight increment value (the initial value given is 1.0)
- W_T is the total number of weights in the neural network
- $N_n(0,1)$ is a Gaussian random variable with zero mean and unit variance
- $v = 1/(\sqrt{2D})$ and $D = \sqrt{\text{the total number of neural network nodes}}$

Consider the following example for the neural network used in NeuroWarri. The initial values were as follows.

$p = 1.0$
$D = \sqrt{1 \text{ output node} + 40 \text{ hidden nodes} + 84 \text{ input nodes}} = \sqrt{125} = 11.1803398875$
$v = 1/(\sqrt{2D}) = 0.2114742527$

The value v remains constant throughout the evolutionary process. On the first attempt at changing the weights, the new value of p is calculated as follows.

$$p' = p \times \exp(vN(0,1)) = 1.0 \times \exp(0.2114742527 \times N(0,1)) = 1.3717781703$$

This is then used to perturb the weights to give new values, as the sample in table 1 demonstrates. Consider the first row of the table, the evolved weight is calculated as follows.

$$w'_1 = w_1 + p'N_1(0,1) = 0.0872393916 + 1.3717781703 \times N_1(0,1) = 0.1634517530$$

Hence p' and the random value $N_1(0,1)$, together act as the error correction value used in normal neural network training. This equation is applied to each weight in the neural network, resulting in an evolved version of the network.

Initial Weight Values	Evolved Weight Values
0.0872393916	0.1634517530
0.1437231765	-0.0768103592
0.0875561732	-0.0720347883
0.0221265776	-0.0118374838
0.1002390516	0.0705144008

Table 1: Sample Weight Values Before and After an Evolutionary Cycle

After 60 generations, the network with the highest score was chosen. During training, a two-ply minmax tree was used i.e. the system only looked two moves ahead while during actual game play the system looked three moves ahead. In the first case, the system picked moves that minimised the opponent's gain while in the second the system picked moves that maximised its own position. In spite of the small minmax tree and the short training time, a reasonable Warri game playing system was produced.

During training, the only information provided to the NeuroWarri system was,

- The objective was to gain more seeds than the opponent.
- A house with zero seeds could not be selected.
- Only the houses on its side could be accessed.

The neural networks then learnt the rules of the game by observing the changes in the state of the board for each move made. The networks then developed specific strategies for achieving the primary objective of obtaining the most seeds. The learning of the rules and the development of the strategies took place at the same time with no visible separation. This makes sense since many of the developed strategies would allow the networks to adhere to the rules.

11.6 Strategies Learnt During Evolution

NeuroWarri developed several game-playing strategies that are similar to the ones human players learn when playing the game.

11.6.1 Attacking Moves

In response to the emphasis placed on gaining the most seeds, NeuroWarri developed strategies for making attacking moves, in order to increase its seed count. NeuroWarri developed two forms of attack: Direct attacks and house building.

11.6.1.1 Direct Attacks

Direct attacks consist of filling the empty houses on the opponent's side in order to launch an attack. It was found that in the early stages of the game, as soon as a house became empty on the opponent's side, NeuroWarri would fill it and then attempt to capture the seeds as shown in the following example. Figures 11 to 14 show the board before and after this maneuver.

```
        X | 0
 f   e   d   c   b   a
------------------------
 4 | 5 | 6 | 6 | 8 | 1
 4 | 0 | 0 | 0 | 7 | 7
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 11: Before Capture (Before Move X.e)

```
        X | 0
 f   e   d   c   b   a
------------------------
 5 | 0 | 6 | 6 | 8 | 1
 5 | 1 | 1 | 1 | 7 | 7
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 12: After the Move X.e, Before the Move Y.A

```
        X | 0
 f   e   d   c   b   a
------------------------
 5 | 0 | 6 | 6 | 8 | 1
 0 | 2 | 2 | 2 | 8 | 8
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 13: After the Move Y.A, Before the Move X.d

```
        X | 9
 f   e   d   c   b   a
------------------------
 6 | 1 | 0 | 6 | 8 | 1
 1 | 0 | 0 | 0 | 8 | 8
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 14: After Capture

In figure 11, NeuroWarri makes the move X.e, which fills the houses, Y.B, Y.C and Y.D (figure 12). The opponent then makes the move Y.A, further exposing its seeds (figure 13). NeuroWarri then plays X.d, resulting in a gain of 9 seeds (figure 14). This strategy in the beginning of the game allowed NeuroWarri to capture pieces early. Later in the game, NeuroWarri would generally switch its strategy to that of building houses, as discussed in the next section.

This strategy also turned out to be its weakness when challenged against more advanced Warri games. These games would move their seeds to avoid capture or allowed NeuroWarri to capture the pieces, placing the board in a state that was disadvantageous to NeuroWarri. NeuroWarri's inability to detect this form of attack is due in part to its small search tree.

It should be noted that NeuroWarri will also attack an exposed seed on the opponent's side. Figure 15 shows the board before the attack. Notice that the seed in Y.C is exposed. NeuroWarri plays X.c, capturing the seed (see figure 16), resulting in a gain of 2 seeds.

```
          X | 0
 f   e   d   c   b   a
------------------------
 5 | 5 | 6 | 6 | 8 | 0
 4 | 0 | 1 | 6 | 6 | 1
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 15: Before Capture (Before Move X.c)

```
          X | 2
 f   e   d   c   b   a
------------------------
 6 | 6 | 7 | 0 | 8 | 0
 5 | 1 | 0 | 6 | 6 | 1
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 16: After Capture

Finally, NeuroWarri would generally avoid making any of these attacks if it resulted in the exposure of any of its seeds to an attack from the opponent as shown in figures 23 and 24.

11.6.1.2 Building Houses

NeuroWarri generally attempted to build-up the total seeds in at least two of its houses. It then used one of the houses to fill the empty houses on the opponent's side (as discussed previously), followed by the other large house to perform the capturing. Figure 17 shows the two main houses X.f and X.c. NeuroWarri first attacks with X.f setting up the board in figure 18. Then the moves Y.A, X.d, Y.E, X.b and Y.C are made. After this series of preparatory moves, the X.c house now has 15 seeds as shown in figure 19. The attack is then made by playing X.c, resulting in the board in figure 20. This results in an overall gain of 3 seeds. Notice that after this play, none of NeuroWarri's seeds are exposed to any attack by the opponent Y.

```
          X | 4
 f   e   d   c   b   a
------------------------
11 | 4 | 3 | 13 | 1 | 2
 0 | 0 | 0 | 10 | 0 | 0
------------------------
```

```
 A   B   C   D   E   F
         Y | 0
```

Figure 17: Before Capture (Before Move X.f)

```
           X | 4
 f   e   d   c   b   a
------------------------
0 | 5 | 4 | 14 | 2 | 3
1 | 1 | 1 | 11 | 1 | 1
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 18: After the Move X.f, Before the Move Y.A

```
           X | 7
 f   e   d   c   b   a
------------------------
1 | 6 | 1 | 15 | 0 | 3
1 | 0 | 0 | 12 | 0 | 2
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 19: After the Move Y.C, Before the Move X.c

```
           X | 10
 f   e   d   c   b   a
------------------------
3 | 8 | 3 | 0  | 1 | 4
0 | 1 | 1 | 13 | 1 | 3
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 20: After Capture

Another strategy was to build up a house so that it had enough seeds to traverse the entire board with the last seed falling on the opponent’s side, accomplishing the same as using two houses. The actual choosing of one of these two strategies by NeuroWarri seemed to depend on which provided the earliest opportunity to attack and capture pieces. Figure 21 shows the board before the attack is made from X.e. Notice that the opponent has no seeds left and so NeuroWarri must place seeds in the opponent’s house or suffer the penalty described earlier. NeuroWarri avoids the penalty and captures seeds by playing X.e to give the board in figure 22.

```
           X | 15
 f   e   d   c   b   a
------------------------
6 | 13 | 3 | 3 | 1 | 7
 0 | 0 | 0 | 0 | 0 | 0
------------------------
```

```
 A   B   C   D   E   F
         Y | 0
```

Figure 21: Before Capture (Before the Move X.e)

```
         X | 17
 f   e   d   c   b   a
------------------------
 8 | 0 | 4 | 4 | 2 | 8
 0 | 1 | 1 | 1 | 1 | 1
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 22: After Capture

11.6.2 Defensive Moves

NeuroWarri also developed several defensive moves that were used to protect its seeds from an opponent’s attack.

11.6.2.1 Avoiding Penalties

NeuroWarri always avoided making moves that would leave the opponent unable to make a move. It would delay or abandon an attack in order to avoid this situation. This is a result of the bias to the output discussed in section 4 where the system always tries to maximise its score and minimise that of the opponent. Consider figures 21and 22, not only was NeuroWarri able to gain 2 seeds but it was also able to avoid the penalty at the same time. However, NeuroWarri has been known to make such a move and suffer the penalty, for those situations where such a move would gain enough seeds for it to declare a win.

11.6.2.2 Avoiding Capture of Exposed Seeds

If during the course of the game, any of NeuroWarri’s seeds became exposed, it would move those seeds to a safer house. In figure 23, the seed in X.a can be captured by the move Y.F. To circumvent this, NeuroWarri moves the seed to X.b (see figure 24).

```
         X | 0
 f   e   d   c   b   a
------------------------
 5 | 5 | 6 | 6 | 6 | 1
 4 | 4 | 0 | 5 | 5 | 1
------------------------
 A   B   C   D   E   F
         Y | 0
```

Figure 23: Before Defensive Move (Before the Move X.a)

```
            X | 0
 f    e    d    c    b    a
------------------------
 5 | 5 | 6 | 6 | 7 | 0
 4 | 4 | 0 | 5 | 5 | 1
------------------------
 A    B    C    D    E    F
           Y | 0
```

Figure 24: After Defensive Move

Generally, it was found that NeuroWarri always gave defense of its seeds a higher priority than the capturing of the opponent's seeds. This is again probably due to the feedback of the total seeds for itself and the opponent to the neural network output (see figure 9).

11.6.3 Overall Strategy

Overall, it seems that NeuroWarri has developed a strategy which consists of capturing seeds but never at the detriment of losing any of its own. The attacks consisted of seizing exposed seeds on the opponent's side while building up houses so as to launch a significant attack. While this makes it difficult for beginners to capture seeds, it can be a boon for advanced players who can take advantage of NeuroWarri's unwillingness to sacrifice seeds in order to gain an advantage. This was especially noticeable when it was tested against other computer-based Warri games (see section 7).

11.7 Testing

Figure 25 shows the actual NeuroWarri game. It was written in C++ so as to create the fastest executable possible. There are two main panels: The left panel displays the current state of the Warri board while the right panel contains the current score. To facilitate testing, NeuroWarri automatically saves the current state of the board before exiting. This enabled the player to return to the game at a later date. The system also allows the human opponent to determine who makes the first move. It should be noted that since NeuroWarri has no knowledge of a start game, it always makes the same initial move when it plays first. Testing consisted of pitting NeuroWarri against both human and computer-based opponents, as discussed in the next sections.

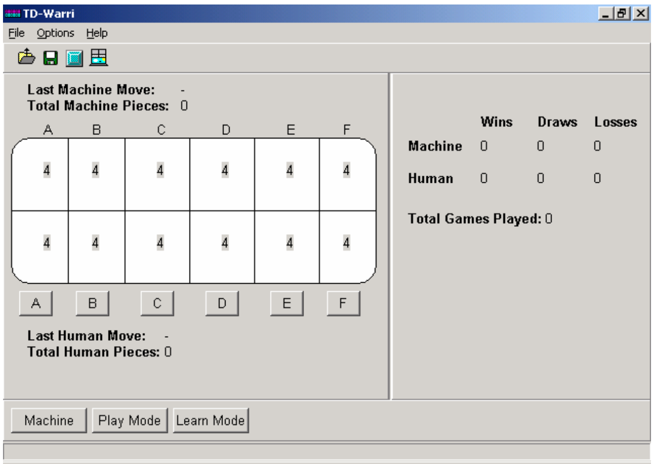


Figure 25: The NeuroWarri User Interface

11.7.1 Human Opponents

Informal testing was performed with both beginners and intermediate players. A beginner was considered to be a person who was playing the game for the first time or had less than 3 months experience. An intermediate player was one with 3 to 12 months of experience. Both groups were handled quite well with the system never losing a game to a beginner and winning the majority of games against the intermediate players. This also included the author of this paper who has won a total of 2 games over a period of a year. Unfortunately, due to the decline in the number of active Warri players in Barbados, expert players could not be located. However, judging by the system's lack of success against more advanced Warri systems, it is expected that it would have had little success against such players.

11.7.2 Other Systems

NeuroWarri was tested against some of the games listed on the University of Alberta Awari web site [5]. It generally performed well at the beginner levels but as the difficulty level of the opposing game was increased, NeuroWarri performance declined until it eventually lost all of the games. The main problem seems to be the relatively low depth of its search tree where NeuroWarri only looks ahead 3 plays. This limits its ability to develop an adequate strategy especially against those games whose search depth increases with the level of difficulty. During several of the games, NeuroWarri was unable to change its strategy even in those cases where the opponent was setting it up for a devastating play.

One advantage that NeuroWarri does have is its short response time. It always responded to an opponent's move in less than 1 second. As a result, NeuroWarri can be made to pause after an opponent has made a move. This enables the human players to study the state of the board before NeuroWarri responds to their moves.

11.8 Future Work

In its current form, NeuroWarri has the skill level of an advanced beginner. To improve this situation, NeuroWarri will be retrained using depths greater than 3. This will allow the exploration of the effects of using greater depths during game play. Also, the ability of the game to vary its depth during game play will be explored.

During the training of the neural networks, only the strongest neural networks were chosen. This may have caused a bias in the genetic material used during the evolution of the networks. Consequently, further research is needed into the effect of letting some of the weaker neural networks generate children so as to increase the genetic spread.

The losses suffered by NeuroWarri when played against other computerised versions of Warri demonstrates that information other than the current state of the board is necessary to win. For example, a start and end game database and the ability to lose seeds in an attempt to set up the board for a grand play.

11.9 Conclusion

NeuroWarri has demonstrated that it is possible to create intelligent computer-based Warri games without prior knowledge of the rules of the game. Evolutionary computation was ideal since it removed the need to create a large set of test patterns with which to train the neural network. Using evolutionary computation, the neural networks can be trained in such a manner that both the rules and strategies are learnt and developed at the same time.

NeuroWarri also demonstrates that it is possible to play a reasonable game of Warri without having to look more than 3 moves ahead. This is in keeping with human nature where most people generally do not look that far ahead. Also it quick response time is in keeping with those exhibited by actual human players.

In conclusion, evolutionary computation has proven to be a strong method for training neural networks in situations were there is no prior information.

11.10 References

[1] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*, Prentice Hall, 1994.

[2] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2nd Edition, IEEE Press, 2000.

[3] D. B. Fogel, "What is evolutionary computation?", *IEEE Spectrum*, Vol. 37, No. 2, Feb. 2000, pp 26-32.

[4] V. W. Porto, D. B. Fogel. and L. J. Fogel, "Alternative neural network training methods", *IEEE Expert*, Vol. 10, No. 3, June 1995, pp 16-22.

[5] The University of Alberta Computer Awari Group. http://www.cs.ualberta.ca/~awari/index.htm.

[6] Elliot Avedon Museum, University of Waterloo. http://www.ahs.uwaterloo.ca/~museum/countcap/pages/warri.html.

[7] A Barbados Web Site. http://barbadosphotogallery.com/warri/index.htm

[8] Mind Sports Olympiad. http://www.msoworld.com/Olympiad/index.html.

[9] Colin Depradine, "NeuroWarri: Using Evolutionary Computation to Develop Warri Strategies", Caribbean Journal of Mathematical & Computing Sciences, University of the West Indies, Cave Hill, Barbados, Vol. 11, 2001, http://scitec.uwichill.edu.bb/cmp/journal/2001_2.doc,

Chapter 12

An Investigation, using Co-Evolution, to Evolve an Awari Player

James Edward Davis and Graham Kendall
Automated Scheduling, Optimisation and Planning Research Group
School of Computer Science and IT, Jubilee Campus, University of Nottingham, Nottingham, NG8 1BB, UK
gxk@cs.nott.ac.uk
http://www.cs.nott.ac.uk/~gxk

Abstract – Awari is a two-player game of perfect information, played using 12 “pits” and 48 seeds or stones. The aim is for one player to capture more than half the seeds. In this work we show how an awari player can be evolved using a co-evolutionary approach where computer players play against one another, with the strongest players surviving and being mutated using an evolutionary strategy (ES). The players are represented using a simple evaluation function, representing the current game state, with each term of the function having a weight which is evolved using the ES. The output of the evaluation function is used in a mini-max search. We play the best evolved player against one of the strongest shareware programs (Awale) and are able to defeat the program at three of its four levels of play.

12 Introduction

Game playing has a long history within AI research. Chess has received particular interest culminating in Deep Blue beating Kasparov in May 1997, albeit with specialised hardware [1] and brute force search, which managed to search up to 200 million positions per second. However, chess is still receiving research interest as scientists turn to learning techniques that allow a computer to ‘learn’ how to play chess, rather than being ‘told’ how it should play [2].

Learning techniques were being used for checkers (draughts) as far back as the 1950’s with Samuel’s seminal work ([3], re-produced in [4]). This would lead to Jonathan Schaeffer developing Chinook, which won the world checkers title in 1994 [5]. Like Deep Blue the question of whether Chinook used AI techniques or not is open to debate. Chinook had an opening and end game database and in certain games it was able to play the entire game from these two databases. If this could not be achieved, a form of mini-max search, with alpha-beta pruning was used. Despite Chinook becoming the world champion, the search has continued for a checkers player that is built using “true” AI techniques. Chellapilla and Fogel ([6],[7],[8]) developed Anaconda, so named, due to the strangle hold it places on its opponent. It is also called Blondie24 [8] which was a name given to the program late in its life in an experiment to see if the name affected the types of player it would attract when playing over the internet and if the other players would treat it differently to a program named something like ‘David0203’. Anaconda (Blondie24) uses an artificial neural network (ANN), with 5046 weights, which are evolved via an evolutionary strategy. The inputs to the ANN are the current board state, presented in a variety of spatial forms. The output from the ANN is a value which is used in a mini-max search. During the training period the program is given no information other than a value which indicates how it performed in the last five games. It does not know which of those games it won or lost, nor does it know if it is better to achieve a higher or a lower score. Anaconda is certainly not given any strategy and contains no database of opening and ending game positions. The aim was to develop a game playing program that has no knowledge of the game, other than how to play

legally, and to show that it can evolve its own strategies. Co-evolution is used to develop Anaconda, by playing games against itself. Anaconda has achieved expert levels of play (ratings of over
2000)

Both checkers and chess are games of perfect information, otherwise known as combinatorial games [9]. These games are classified as two-player games, with no hidden information, no chance moves, a restricted outcome (win, lose and draw) and with each player moving alternately. This is different to games such as poker [10], [11], [12], backgammon [13], [14], [15], [16], or bridge [17], [18], [19], [20] where there is hidden information, a chance element and, possibly, more than two players. A recent survey of computers and game playing [21], covers those games above, as well as others.

In this work we look at a combinatorial game called (amongst other things) Awari. We show how a coevolutionary approach can produce a player that can play to a reasonably high level. There have been limited studies reported in literature that have looked at the game of Awari, although there are many Awari programs and competitions. The strongest shareware program is believed to be Awale by Myriad Software (http://www.myriadonline. com/awale.htm) and this is the program we use to test our developed player.

Bambam, from the research group at the University of Alberta led by Jonathan Schaeffer, was created in May 1999 to compete in the Awari tournament at the same university. It had a short term aim of becoming the strongest player on the planet and a long term aim of solving the game. The name, Bambam, was used due to its brute force play and the fact that it played with pebbles. One of the strongest players is Lithidion, produced in 1990 by Maarten van der Meulen and Victor Allis to compete in the Computer Olympiads. It won the gold medal in 1990, 1991, and 1992, at which time it was retired. Since that time there have not been any other serious computer tournaments so nothing has been able to challenge Lithidion. Thomas Lincke is working on solving the game and has produced an endgame database up to 37 seeds. His web site (http://www.shortestwin.com/awari/) allows you to interactively explore these positions.

12.1 Awari

The game of Awari is one of the oldest known strategy games. It is believed to have originated in Ethiopia about 3500 years ago and has since spread across Africa. The game is known by many different names, for example, Awale, Awele, Ajwa, Lela, Chisolo, Kalak, Oware, Coo, Coro Bawo, Nocholokoto, Dara, Congkak, Mancala, Bawo, Omweeso, Adita-ta, Kasonko, Layo, Gilberta, Schach, Wari and Walle; and this list is by no means complete. Awari is played with a hollowed out plank of wood and a number of stones or seeds.

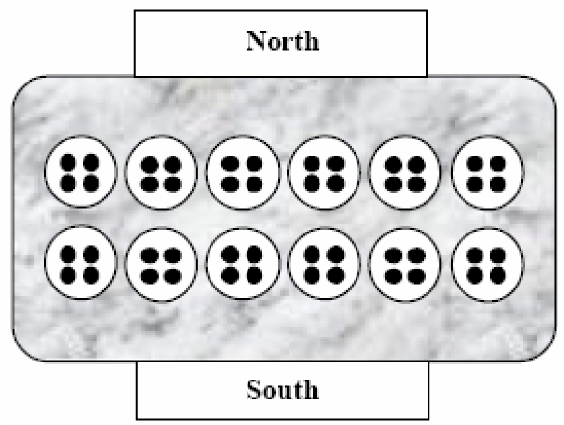


Figure 1: Initial setup for Awari board

The plank has twelve hollows (pits), six designated as belonging to North and six belonging to South. At the start of the game, each pit contains four seeds (figure 1). As the game progresses each pit can contain any number of seeds, although the total number of seeds remains constant (48). The aim of the game is to capture the majority (>24) of the seeds. Once a seed is captured it is removed from the board and plays no further part, other than being used to evaluate the current game position. The game proceeds as follows. South plays first and picks up all the stones from a nonempty pit on his/her side of the board. He/she deposits the stones, in an anti-clockwise direction, dropping one stone in each pit until all the stones have been deposited. If the number of seeds means that the player passes completely around the board, the starting pit is skipped. If the final stone is dropped in one of the opponents pits which has two or three stones in it (after the deposit), then those stones are captured. If the proceeding pit now has two or three stones in it, these stones are also captured. Capturing continues, in a clockwise direction, until either a pit does not have two or three stones in it, or the pit under consideration is one of your own pits. Figure 2 shows an example play, where both players can capture stones.

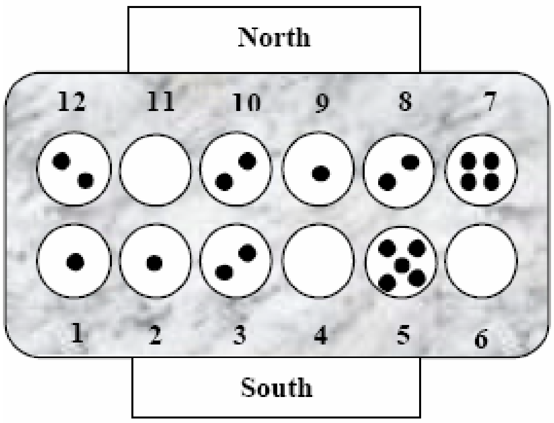


Figure 2 : Example play from Awari; south to play

If South plays pit five it will deposit seeds in pits 6, 7, 8, 9 and 10. It will capture the seeds in pits 10, 9 and 8 (total of 8 seeds captured). If North now plays pit 12 it will deposit seeds in pits 1 and 2 and will capture 4 seeds.

Figure 3 shows a typical position that players try to manoeuvre towards, as part of the strategy of the game. This is known as a Kroo, that is the accumulation of more than 12 seeds in one pit to allow for a complete revolution of the board. It also shows another strategy; the starvation of a players pits so that they are limited in the number of moves they can make. Figure 3 explains why a Kroo is such an important strategy.

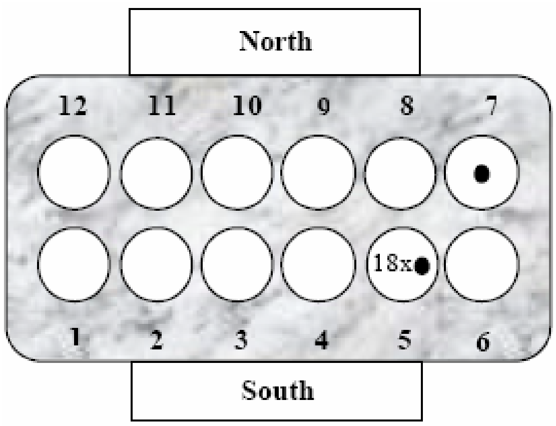


Figure 3 : Awari Strategy

When South plays pit 5 it will finish depositing in pit 12 (remember that the starting pit is skipped if play passes around the board). As play has passed around the board all the opponents pits will contain two stones, with the exception of pit 7 which will contain 3 stones. This results in all the seeds on the North side being captured (a total of 13 seeds). Awari has as many rule variations as it has names. The rules we used in this study are given in appendix A.

12.2 Evolving an Awari Player

The aim of this study is to develop a player that initially plays a poor (random) game of Awari but is able to evolve to play a better game as the player evolves using a coevolutionary approach. The work of Fogel [8] used a neural network to determine a value that represented the current board position. This value was used in a mini-max search to decide which move the computer player should make. In effect the neural network represented a function that returned the current game state at a given time. In this work we are presenting the computer player with a simple evaluation function to ascertain if co-evolution is able to optimise the function to a level where the player can player Awari to a sufficiently high level. The function we present to the evolving player is as follows

$$f = w_1 a_2 + w_2 a_3 + w_3 \beta_2 + w_4 \beta_3 + w_5 a_s + w_6 \beta_s \qquad (1)$$

where:

$w_1..w_6$ The weightings for each term of *f*
a_2 The number of the opponents pits vulnerable to having 2 stones captured on the next move
a_3 The number of the opponents pits vulnerable to having 3 stones captured on the next move
β_2 The number of the evolving players pits vulnerable to having 2 stones captured on the next move
β_3 The number of the evolving players pits vulnerable to having 3 stones captured on the next move
a_s The current score of the opponent
β_s The current score of the evolving player

This function was an initial attempt to devise a set of terms that seemed likely to capture the important elements of the game. Whether this function would allow an evolved player to improve its play over time was not known at the time the function was derived. Our aim was to see if, given a simple evaluation function, the player could evolve a good strategy. It is the weights for each term ($w_1..w_6$) that are evolved using a co-evolutionary strategy. The evolutionary process is conducted as follows:

- A population, P, of 20 players is created. Each member of the population, p*n* (*n*=1..20), contains six real numbers which correspond to the weights, *w*1..*w*6. The weights are randomly initialised with values –1..+1.
- Each pn plays every other pn member twice, but they do not play themselves. They play once as north and once as south.
- For each move by the evolving player, a search tree is constructed. The depth of the search tree is determined by the available search time. In our experiments the search depth was 7. This was chosen after experimentation as a good trade off between the search needed by the player and the time taken to build the search tree. At this depth the search took about one minute. The evaluation function (1) assigns a value to each of the terminal nodes and these values are propagated up to the root of the search tree using the mini-max algorithm. The value at the root is used to decide which move to make.
- The winning player is awarded 3 points for a win, 1 point for a draw and zero points for a loss.
- If a game reaches move 250 the points are awarded on the state of the game at that point. There is an argument for simply awarding a draw but we decided against this and awarded a winning score for capturing more seeds.
- At the end of all the games the top *m* (for our experiments *m*=5) players were retained and the rest were discarded.
- Each retained player produces an equal number of children. If this would exceed the population size (as *n*=20 and *m*=5, this was not relevant to us) then the production is biased towards the fittest individuals. The production of a new player is produced by

$$pn(w_i) = pn(w_i) + N(0,1) \qquad (2)$$

where the standard normal variable is sampled anew for each weight (see [22] for sample code to produce normal variables). This method of adaptation was used to mimic the successful approach by Fogel [8].

- We ran 250 generations in order to produce our evolved player.

12.3 Results

To compare our evolved player we used the game of Awale produced by Myriad Software. The shareware version of this package allows one level of play (initiation). If you register the software you are given access to three higher levels (beginner, amateur and grand master). Myriad were kind enough to supply us with the registered version for the purposes of this research, so that we could test our evolved player against all levels. Initially we tested a random set of weights against Awale at the lowest level (table 1). This player, not surprisingly was easily beaten by Awale. The game finished after 68 moves when Awale had captured 26 seeds. We also played the same random player at the grand master level (table 2) and the player was easily defeated after 50 moves when Awale had captured 29 seeds.

TABLE 1
PLAYING AWALE AT INITIATION LEVEL WITH RANDOM WEIGHTS

Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
68	0	26

TABLE 2
PLAYING AWALE AT GRAND MASTER LEVEL WITH RANDOM WIEGHTS

Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
50	0	29

The best evolved player from the algorithm presented above was played at each level of Awale over a series of five games. The first set of games is presented in table 3, when the evolved player competed against Awale as its lowest playing level (initiation). The results show that Awale was easily defeated by the evolved player. On average, the evolved player captured the majority of seeds in just over 47 moves. The numbers of seeds captured by Awale averaged just under 3.

TABLE 3
PLAYING AWALE AT INITIATION LEVEL

Game #	Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
1	29	30	3
2	39	32	0
3	39	32	7
4	67	27	2
5	63	28	2
	47.40	**29.80**	**2.80**

Table 4 shows the results when playing against the beginner level of Awale. Again, the evolved player was a relatively easily winner although the average number of moves has increased as has the average number of seeds captured by Awale. The fourth game of this series is interesting as it appears as if the evolved player has not captured the majority of the seeds. In fact, it won due to the fact that it captured all the stones due to a play it made (see rule 7, in appendix A). The sequence final sequence that led to this position is described in Appendix B.

TABLE 4
PLAYING AWALE AT BEGINNER LEVEL

Game #	Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
1	43	30	8
2	75	28	6
3	43	25	4
4	33	19	9
5	85	29	12
	55.80	**26.20**	**7.80**

Table 5 shows the five game series against Awale's amateur level. Things are getting a little tougher now. The average number of moves has risen dramatically and the evolved player suffered its first defeat (game 4). There was also a draw in this series (game 3) where the game was stalemated in that the same position kept repeating itself. In fairness to Awale, we did note that it could have made a winning play on a number of occasions but it did not exploit it. Not surprisingly, the average number of seeds captured by Awale has also risen dramatically in this five game series. One of the games from this series can be seen in appendices C. In particular, it shows that, although the evolved player won it does not does value winning as highly as capturing.

TABLE 5
PLAYING AWALE AT AMATEUR LEVEL

Game #	Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
1	59	35	6
2	103	32	9
3	158	15(d)	24(d)
4	116	15	26
5	105	24	19
	108.20	**24.20**	**16.80**

Table 6 shows the results from playing against Awale at its highest level (grand master) and unfortunately it shows that our evolved player is no match for Awale when playing at this level.

When playing at this level Awale asks you to set a maximum time allowed for each move and also an analysis depth. We set the move time 60 seconds (to match with our search time) and the depth at the lowest possible value, 12. However, we have no information as to what the analysis depth relates to. We cannot assume, for example, it is the search depth of a mini-max search. In addition, we have no information about the evaluation function used by Awale but, we suspect, it is more sophisticated that the one evolved by our player.

TABLE 6
PLAYING AWALE AT GRAND MASTER LEVEL

Game #	Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
1	86	4	25
2	76	4	28
3	86	4	25
4	76	5	28
5	76	5	28
	80.00	**4.40**	**26.80**

12.4 Discussion

The purpose of this study was to see if, presented with a simple, certainly not optimised, evaluation function a game could evolve that improved its play from an initial, random state. We have shown that this is possible and that the evolved player achieves reasonable level of play (certainly able to beat the authors). Future work will look at developing a player that is able to find its own objective function and its own strategies. The work of Fogel [8] already provides inspiration for this but there is also scope for using other machine learning techniques (such as genetic programming) to evolve a suitable evaluation function.
We would like to think that ultimately we could challenge the best human players.

12.5 Acknowledgements

The authors would like to express their thanks to Myriad Software, who supplied us with a licensed version of their Awale program and also to the reviewers for their helpful comments.

Appendix A : Rules of Awari

These are the rules that were followed for this study. There are many variations, many of which can be defined within the Awari program that we used (Awale).

1. The board comprises 12 pits (labeled 1..12), each with four seeds. Six of the pits belong to south (1..6). The other six belong to north (7..12).
2. The game starts with the players selecting who is North and who is South. South moves first.
3. On your turn, select a non-empty pit on your side of the board. "Sow" the seeds from that pit around the board, dropping one at a time, counter-clockwise into each pit.
4. If you choose a pit with enough seeds to go completely around the board (12 or more), the original pit is skipped and left empty.
5. If the last seed is dropped into a pit on your opponent's side, leaving that pit with 2 or 3 seeds, you capture all the seeds in that pit. The capture continues with consecutive ***previous*** pits on that side which also contain 2 or 3 seeds.
6. If all your opponent's pits are empty, you must make a move that will give him a move. If no such move can be made, you capture all the remaining seeds on the board, ending the game. If no move is possible, the winner is the person with the greater number of captured seeds.
7. If, by making a play, you can capture all the stones on your opponent’s side of the board, you win the game (as your opponent cannot make a play). Note, this overrides rule 6.
8. At the end of the game the seeds left on the board are not captured by any player.
9. The game is over when one player has captured 25 or more seeds, or both players have taken 24 seeds each (a draw).

Appendix B : End Play Against Beginner Level of Awale (Game 4)

When the evolved player played Awale at the beginner level the game ended when the evolved player captured all the stones on the opponents side (see rule 7 in appendix A). This is the final sequence that led to that position. Awale is playing north and is next to play. The scores at this time are north=9, south =10. North must play the pit with 15 seeds.

North

15	0	0	0	0	0
0	0	10	0	5	2

South

This leads to this position

North

0	1	1	1	1	1
2	2	12	2	6	3

South

When south plays it decides to play the pit with 6 seeds. This leads to the position shown below, where south captures 10 stones but, more importantly, does not give north a move and thus wins the game.

North

0	0	0	0	0	0
2	2	12	2	0	4

South

Although the play above may seems obvious, the real credit goes to the evolved player in that it created this position in order to exploit it.

Appendix C : End Play Against Amateur Level of Awale (Game 2)

This is the final sequence of game 2 when the evolved player played Awale at the amateur level. Awale is playing north. South (the evolved player) is next to play. The scores at this time are north=9, south=19 and the current position is as follows.

North

0	1	0	0	0	0
0	0	0	2	17	0

South

South plays the pit with 2 seeds forcing north to play its pit with a single seed, leading to this position.

North

1	0	0	0	0	0
0	0	0	0	18	1

South

The evolved player decides to play the pit with 18 seeds, leading to this position (before the capture).

North

3(*)	2	2	2	2	2
1	1	1	1	0	3

South

The capture starts at the pit marked with an asteric, leading to a capture of 13 seeds, giving south a total of 32 seeds, and the game. Notice that from the initial position shown in this play south could have won immediately by playing the pit with 17 seeds. However, this would have only captured 11 seeds. This suggests that the evolved player gives more importance to capture than to winning the game. Whilst some human players may play like this (to inflict even more humiliation on their opponent) it would probably normally be better to secure the win as soon as possible.

12.6 References

[1] S. Hamilton, L. Garber, "Deep Blue's hardware-software synergy". IEEE Computer, 30(10), pp 29-35, October 1997

[2] G. Kendall, G. Whitwell "An Evolutionary Approach for the Tuning of a Chess Evaluation Function using Population Dynamics". In proceedings of CEC2001 (Congress on Evolutionary Computation 2001), COEX Center, Seoul, Korea, May 27-29, pp 995-1002, 2001

[3] A. L. Samuel "Some Studies in Machine Learning using the Game of Checkers". IBM Journal of Research and Development, 3(3), pp 210- 229, 1959

[4] A. L. Samuel "Some Studies in Machine Learning using the Game of Checkers". IBM Journal of Research and Development, Vol. 44 No. 1/2 January/March, pp 207-226, 2000

[5] J. Schaeffer, R. Lake, P. Lu "CHINOOK The World Man-Machine Checkers Champion", AI Magazine, 17(1), pp 21-30, 1996

[6] K. Chellipilla, D. B. Fogel, "Evolution, Neural Networks, Games, and Intelligence." Proc. IEEE, Vol 87, No. 9, pp 1471-1498, 1999

[7] K. Chellipilla, D. B. Fogel "Anaconda Defeats Hoyle 6-0: A Case Study Competing an Evolved Checkers Program against Commercially Available Software." In Proceedings of Congress on Evolutionary Computation, July 16-19 2000, La Jolla Marriot Hotel, La Jolla, California, USA, pp 857-863, 2000

[8] D. B. Fogel, "Blondie24 : Playing at the Edge of AI". Morgan Kaufmann, 2001, ISBN:1558607838

[9] A. S. Fraenkel, "Combinatorial Games : Selected Bibliography with a Succinct Gourmet Introduction", Games of No Chance, Vol. 29, MSRI Publications, Cambridge University Press (ed. R. Nowakowski), pp 493-537, 1996

[10] G. Kendall, M. Willdig, "An Investigation of an Adaptive Poker player". The 14th Australian Joint Conference on Artificial Intelligence (AI'01), Adelaide, Australia, 10 - 14 December 2001

[11] L. Barone, L. While, "Adaptive Learning for Poker". In proceedings of the Genetic and Evolutionary Computation Conference, pp 566 - 573, 2000

[12] J. Schaeffer, D. Billings L. Peña, D. Szafron, "Learning to Play Strong Poker." In proceedings of the Sixteenth International Conference on Machine Learning (ICML-99), J. Stefan Institute, Slovenia (Invited Paper), 1999

[13] P. J. Darwen, "Why Co-Evolution beats Temporal Difference Learning at Backgammon for a Linear Architecture, but not a Non-Linear Architecture." In proceedings of CEC2001 (Congress on Evolutionary Computation 2001), COEX Center, Seoul, Korea, May 27-29, pp 1003-1010, 2001

[14] B. Pollack, A. D. Blair, "Co -evolution in the Successful Learning of Backgammon Strategy." Machine Learning, 32(3), pp 225-240, 1998

[15] G. Tesauro, "Temporal Difference Learning and TD-Gammon." Communications of the ACM, 38(3), pp 58-68, 1995

[16] G. Tesauro, "Comments on Co-Evolution in the Successful Learning of Backgammon Strategy." Machine Learning, 32(3), pp 241-243, 1998

[17] L. Frank, "Search and Planning under Incomplete Information: A Study Using Bridge Card Play." Springer Verlag, 1998

[18] M. Ginsberg, "GIB: Steps Toward an Expert-level Bridge-Playing Program." In proceedings of International Joint Conference on Artificial Intelligence, pp 584-589, 1999

[19] S. Smith, D. Nau, T. Throop, "Computer Bridge: A Big Win for AI Planning." AI Magazine, 19(2), pp 93-105, 1998

[20] S. Smith, D. Nau, T. Throop, "Success in Spades: Using AI Planning Techniques to Win the World Championship of Computer Bridge." In proceedings of AAAI National Conference, pp 1079-1086, 1998

[21] J. Schaeffer, "The Games Computers (and People) Play." Academic Press, Vol. 50 (ed Zelkowitz M.V.), pp 189-266, 2000

[22] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1996). Numerical Recipes in C, 2nd ed., Cambridge University Press, ISBN 0-521-43108-5

Chapter 13

Solving an Ancient African Game

Ivars Peterson

13 Introduction

The game of awari has entranced players for thousands of years. Originating in Africa, it remains a popular pastime in many parts of the world. Awari and its numerous variants are instances of "count-and-capture" strategy games, and they are known generically as *mancala* games.

In its traditional form, the awari game "board" consists of two parallel rows of six hollows, with four markers (beans) in each hollow, or cup. Two players sit opposite each other, with six cups belonging to one player and six to the other. The game's goal is to capture the most beans.

Player 2	4	4	4	4	4	4
Player 1	4	4	4	4	4	4

13.1 Awari: initial configuration

The first player takes all the beans from any one of the six cups on his or her side and, moving anticlockwise, adds one bean to each succeeding cup, until all the beans are used up. The second player then takes the beans from one of the six cups on his or her side and does the same.

Player 2	4	4	4	4	4	5
Player 1	4	4	0	5	5	5

Above: Configuration after player 1 takes a turn.

Player 2	5	0	4	4	4	5
Player 1	5	5	1	5	5	5

Above: Configuration after player 2 takes a turn.

When a player drops his or her last bean into a cup on the opponent's side containing only one or two beans (making a total of two or three beans), that player removes all the beans from this cup, taking them out of the game. The same player also takes any beans in cups immediately before the emptied cup if they now also total two or three.

		Player 2			
Cup 12	Cup 11	Cup 10	Cup 9	Cup 8	Cup 7
2	0	2	1	2	4
Cup 1	Cup 2	Cup 3	Cup 4	Cup 5	Cup 6
1	1	2	0	5	0
Player 1					

If player 1 plays the five beans in the fifth cup, beans are deposited in cups 6, 7, 8, 9, and 10. The player captures the beans in cups 10, 9, and 8 (a total of eight beans). If player 2 then plays the two beans in cup 12, beans go into cups 1 and 2. The player captures the beans in cups 1 and 2 (a total of four beans).

Players take beans only from their opponent's side. As the game progresses, a given cup can contain any number of beans, but the total number (including captured beans) remains 48. The game ends when one player has no beans left in the cups on his or her side and, hence, cannot move any beans. The winner is the player who has captured the majority of the beans.

Technically, awari is a two-person game of perfect information. Like chess, checkers, and other games in which chance is not involved, it has been of particular interest to researchers in the field of artificial intelligence. Over the years, programmers have also created software to play the game. A number of these computer programs faced off against each other in a computer awari tournament, held in 1999 at the University of Alberta. At that time, no program could guarantee a win or even a draw.

To figure out what happens in the game, computer scientists John W. Romein and Henri E. Bal of the Free University in Amsterdam wrote a computer program that calculates the best move and eventual outcome for all 889,063,398,406 positions that can possibly occur in the game.

Their result? As in tic-tac-toe, when you and your opponent play perfectly, the game always ends in a draw. A computational tour de force, the calculations required about 51 hours on a large computer cluster with 144 processors, orchestrated by a novel divide-and-conquer algorithm for distributing the computations. The results were stored in a 778-gigabyte database.

Romein and Bal are now developing what they describe as an "invincible" awari program, which uses the database to play perfectly. In the meantime, visitors to their Web site at http://awari.cs.vu.nl/ can play an online version of the game, which is programmed to play at a less-than-optimal level so that a good human player can actually still beat the computer.

13.2 *References*

Davis, J.E., and G. Kendall. Preprint. An investigation, using co-evolution, to evolve an awari player. Available at http://www.cs.nott.ac.uk/~gxk/papers/cec2002gxk.pdf.
The complete rules of awari can be found at http://www.awari.com/ or http://www.cs.ualberta.ca/~awari/rules.html.
You can play awari online at http://awari.cs.vu.nl/. See http://www.myriad-online.com/awalink.htm for links to additional Web sites where you can play the game or download software for playing it on your computer or even cell phone.
The University of Alberta computer awari group has a Web page at http://www.cs.ualberta.ca/~awari/.
****
A collection of Ivars Peterson's early *MathTrek* articles, updated and illustrated, is now available as the MAA book *Mathematical Treks: From Surreal Numbers to Magic Circles*. See http://www.maa.org/pubs/books/mtr.html.
http://www.free-download-game.com/board/board-games-by-name-page4.html
http://www.gamecabinet.com/rules/Bao.html
International Omweso Society: http://www.omweso.org/
http://www.cs.ruu.nl/~hansb/d.gam/mancala.html (games list, software etc)

Chapter 14

Combinatorics in African Board Games: Illustration Using “Ayo”

14 Introduction

Broline and Loeb [1] provide one of the fundamental works and the incentives to apply combinatorics to African board games. Their excellent work his is the foundation for this Chapter. It provides endgame considerations in the two-player game Ayo originating from the Yoruba ethnic group of Nigeria. Although Ayo is the example, techniques covered equally apply to other board games of similar characteristics. Broline and Loeb determined the periodicity of the hole occupancies in s stone winning positions. A winning position is an arrangement of game seeds which lead to a definite win scenario. Given n holes, the number of seeds in a winning position is found to be asymptotically bounded.

There are certain common unbalanced Ayo endgame positions called ***determined positions***. For all s, there is a unique such position with s stones. Of course, certain positions are not realizable on a finite board with a fixed number of holes. It can be shown that the number of stones in such a position on a board with 2n holes is bounded by approximately $\boldsymbol{n}^2/\pi$. Broline and Loeb studied the actual distribution of stones into holes, and discover a periodicity in the contents of the first k holes (with respect to the total number of stones) of $\boldsymbol{lcm}(1;2;\cdots;\boldsymbol{k}+2)$.

14.1 Rules of Ayo

The game Ayo has two rows of six holes each. The rules [2] are as follows:

Set up **-** 48 stones are used. Initially, 4 are placed in each of the 12 holes. (We will generalize the game somewhat allowing boards with 2n holes and an arbitrary placement of stones.)

Players **-**Two players alternate making moves. Each player's side of the board has n holes.

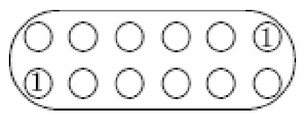
Objective The object of the game is to capture the most stones.

Movement - To move, a player chooses a non-empty hole from his or her side of the board, and removes all of its stones. The stones are redistributed (sown), one per hole, among the holes in a counter clockwise direction beginning with the hole after the chosen hole.

Odu - A hole which contains 2n or more stones is said to be an Odu [2]. If the chosen hole is an Odu, the redistribution proceeds as usual except that the initial hole is skipped on each circuit of the board.

Capture **-** If the last hole sown by a player is on the opponent's side of the board and contains (after having been sown) two or three stones, then the stones in this hole are captured. Also captured are stones in the consecutively preceding holes which meet these conditions.

End of Game **-** At each turn, a player must, if possible, move in such a way that his or her opponent has a legal move. If, on some move, a player cannot move in such a way to give his or her opponent a legal move, the game is over and the player is awarded all remaining stones. If there are so few stones on the board that neither player can ever capture, but both players will always have a legal move, the game is over and each player is awarded the stones on his or her own side of the board. For example, if the position is no further captures are possible, but each player can always move to give the opponent a legal move. In this case, each player is awarded a single stone.



The game opens rapidly with both players showing dexterity and skill by the speed of their movements. However, playing the game well requires remembering the number of stones in each of the twelve holes, as well as planning several moves in advance. Thus, the opening game is both interesting to watch and difficult to learn. The endgame is less exciting, but easier to analyse. The latter stages of the game tend to be dominated by one player. She can move in such a way that her opponent has at all times only one legal move. After this sequence of moves, only a few stones usually remain, and no further captures are possible.

We shall analyse a specific type of endgame on a generalized Ayo board with 2n holes. The holes will be numbered clockwise $-n+2;\ -n+1\ \cdots\ 1;\ 0;\ \cdots\ n-1;\ n;\ n+1$ (Figure 1.). The two players will be denoted S (South) and N (North).

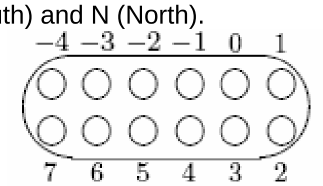
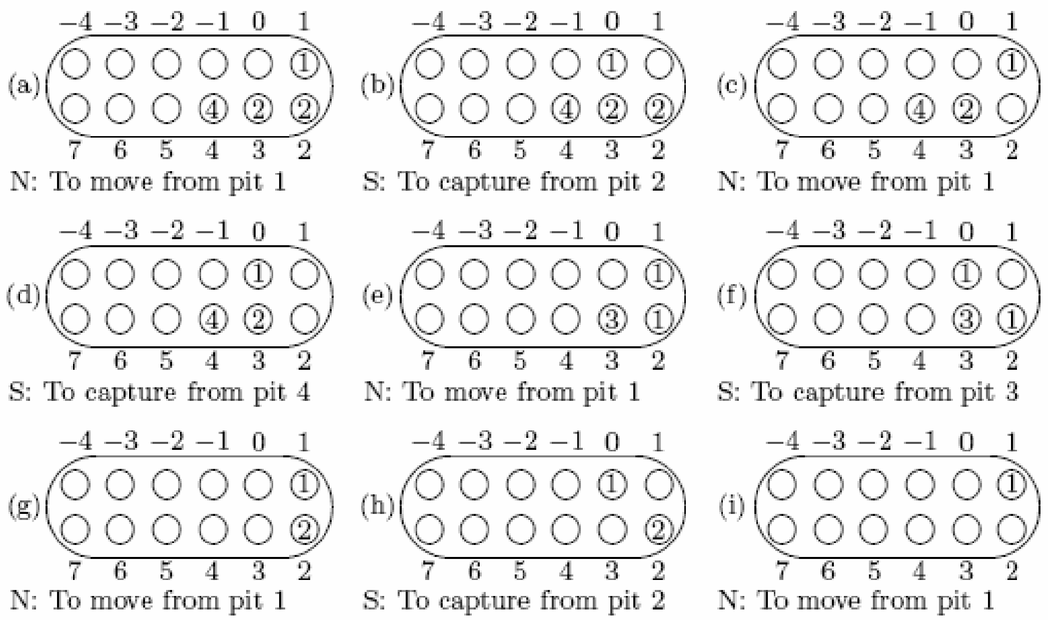


Figure 18: The standard Ayo board numbering

S makes her plays from holes numbered from n+1 down to 2, while N makes his plays from holes numbered from 1 down to _n + 2. Play proceeds from higher numbered holes to lower numbered ones (and from hole _n + 2 to hole n + 1). The endgame positions we shall study are those which satisfy the following definition.

Definition 1: *A determined position is an arrangement of stones on a generalized Ayo board where it is possible for S to move such that*

- *S captures at every turn,*
- *there is no move from an Odu*
- *after every turn, N has only one stone on his side of the board, and*
- *all stones are captured by S except one which is awarded to N*



Note: All captures are to hole 0.

Figure 19: The Determined Position with Nine stones [Example from reference [1]]

Figure 2 shows a determined position and the subsequent play between the two players on a board with 12 holes. Initially, there are nine stones and it is N's turn. Eight stones are captured by S and one is awarded to N. It is possible to show that a determined position on a 12 hole Ayo board has at most 21 stones. It is a simple matter to establish the contents of the holes on N's side of the board in a determined position. The study of the contents of the holes on S's side of the board will be more rewarding.

Lemma 2 *The stone on N's side of a determined Ayo position must be in hole 1 if N is to move, and in hole 0 if S is to move*.

Proof: If S is to move, she must capture and leave only one stone. Thus, the stone captured must lie in N's second hole (hole 0). Hence, before N's move, the stone must have been in hole 1.

There is a simple strategy by which any feasible win can be forced.

Proposition 3 *([3, 4]) If a win is possible from a given Ayo position, the unique winning move must be to harvest the smallest harvestable hole.*

Proof: Suppose holes i and j are both harvestable (i.e., they contain i and j stones, respectively) and i < j. If hole j is harvested, then hole i will contain i+1 stones. It would then be overfull," and could no longer be harvested. Further play could only increase the occupancy of hole i.

This strategy can be used [3, 4] to enumerate a large number of winning positions. In fact, the strategy can be applied backwards. That is to say, given a winning position, one can obtain a winning position with one more stone in the following manner: Let $i \geq 1$, be the least number such that hole i is empty. Place i stones in this hole, and remove one stone from all previous holes. (This can be done since by definition they are non-empty.) Applying the winning strategy involves removing the i stones and sowing back 1 stone into all the holes from which it was removed. We thus obtain an explicit bijection between winning positions with s stones and those with s + 1. Since there is but one position (winning to be sure) with no stones, we have the following result.

Theorem 4 *For all s > 0, there is exactly one winning position involving a total of s stones.*

The winning positions with s < 24 are enumerated in figure 3. Note in particular that at most 21 stones may appear in a determined position on a standard Ayo board. Further calculations were done up to s = 21, 286, 434 using a simple SML program.

Stones s	Pit 1	Pit 2	Pit 3	Pit 4	Pit 5	Pit 6	Pit 7	Pit 8	Harvest h_s
0									
1	1								1
2		2							2
3	1	2							1
4		1	3						3
5	1	1	3						1
6			2	4					4
7	1		2	4					1
8		2	2	4					2
9	1	2	2	4					1
10		1	1	3	5				5
11	1	1	1	3	5				1
12				2	4	6			6
13	1			2	4	6			1
14		2		2	4	6			2
15	1	2		2	4	6			1
16		1	3	2	4	6			3
17	1	1	3	2	4	6			1
18			2	1	3	5	7		7
19	1		2	1	3	5	7		1
20		2	2	1	3	5	7		2
21	1	2	2	1	3	5	7		1
22		1	1		2	4	6	8	8
23	1	1	1		2	4	6	8	1
24				4	2	4	6	8	1

Figure 20: Winning Positions with up to 24 stones

$$\begin{aligned}&\textbf{\textit{fun ayo carry nil}} = [\textbf{\textit{carry}}]\\ &\quad \textbf{\textit{ayo carry}}\ (0 :: \textbf{\textit{xs}}) = \textbf{\textit{carry}} :: \textbf{\textit{xs}}\\ &\quad \textbf{\textit{ayo carry}}\ (\textbf{\textit{x}} :: \textbf{\textit{xs}}) = (\textbf{\textit{x}} - 1) :: (\textbf{\textit{ayo}}(\textbf{\textit{carry}} + 1)\textbf{\textit{xs}})\end{aligned}$$

14.2 Periodicity

The columns of Figure 4 exhibit a certain periodicity. That is to say, the number of stones in the first hole depends not on s but seemingly on s modulo 2. The contents of the first two holes are periodic of period 6 and those of the first three of period 12. An extended table suggests surprisingly that the contents of the first four holes and the contents of the first five holes have the same periodicity, as do the contents of the first eight holes and the contents of the first nine (Figure 4.).

i	1	2	3	4	5	6	7	8	9	10	11
period	2	6	12	60	60	420	840	2520	2520	27720	27720

Figure 21: Period of the contents of the first i holes

This periodicity can be established by an analysis of sequences of numbers determined by the winning positions. Consider the unique winning position with s stones. Let $p_{i,s}$ be the number of stones initially in hole i. Clearly, $p_{i,s} \leq i$.

Let $m_{i,s}$ be the number of times hole i must be harvested in order to win and $b_{i,s}$ be the number of moves of the winning strategy which result in a stone being added to hole i. By convention, $b_{0,s}$ = s. Using hole 0, we thus have $b_{0,s}$ = s. Obviously, for i $\geq$ 1,

$$p_{i,s} = i\, m_{i,s} - b_{i,s} \qquad (1)$$

There is a one to one correspondence between the moves which add a stone to hole I and the moves which harvest some hole j for j>i. Hence

$$b_{i,s} = \sum_{j>s} m_{j,s}$$

It follows that

$$b_{i,s} = m_{i+1,s} + b_{i+1,s} \qquad (2)$$

Proposition 4: For all I, the sequence of i-tuples

$$((p_{1,s}\,;\, p_{2,s}\,;\cdots;\, p_{i,s}))s \geq 0$$

is periodic and has a period $lcm(1, 2, 3, \cdots, i+1)$.

> *Proof:* We show, by induction in I, that not only is the sequence of i-tuples periodic of period $t = lcm(1, 2, 3, \cdots, i+1)$, but also t is the smallest positive number such that
>
> $$p_{j,t} = 0, \quad j = 1, 2, \cdots, i$$

The result is trivial for i = 1. Assume, by induction, that the result holds for all values less than or equal to some i$\geq$1. Let $t = lcm(1, 2, 3, \cdots, i+1)$. The inductive hypotheses imply $p_{j,kt} = 0$, for j = 1, 2, : : : , i and k = 1, 2, : : : : Thus, equations 1 and 2 imply

$$jm_{j,kt} = b_{j,kt} = m_{j+1,kt} + b_{j+1,kt} = (j+2)m_{j+1,kt}\,, \quad j = 1, 2, \cdots, i-1.$$

Combining these results we get

$$2m_{1,kt} = i(i+1)m_{i,kt}\,.$$

Since every other move is an addition to hole 1, $2m_{1,kt} = kt$. Therefore,

$$\begin{aligned} p_{i+1,kt} &= (i+1)m_{i+1,kt} - b_{i+1,kt} \\ &\equiv (i+1)(m_{i+1,kt} + b_{i+1,kt}) \bmod i+2 \\ &= (i+1)b_{i,kt} \\ &= i(i+1)m_{i,kt} \\ &= 2m_{1,kt} \\ &= kt \end{aligned}$$

The smallest positive value of k such that

$$kt \equiv 0 \mod i+2$$

is $k = (i+2)/\gcd(t, i+2)$. Setting q = kt, we have

$$q = ((i+2)t)/\gcd(t, i+2) = lcm(t, i+2)$$
$$= lcm((lcm(1,2,\cdots,i), i+2)) = lcm(1,2,\cdots,i+2).$$

Thus, we have shown

$$p_{j,q} = 0, \quad i = 1,2,\cdots,i+1$$

and q is the smallest positive multiple of t for which this is true. By the inductive assumption, we can thus deduce that q is the smallest positive integer for which this is true. In particular, the sequence of i + 1-tuples

$$((p_{1,s}; p_{2,s}; \cdots; p_{i+1,s}))s \geq 0$$

does not have period less than q.

Now, the contents of the first i+1 holes in the winning position with q stones are the same as the contents of these holes in the winning position with no stones. Suppose for some s that the contents of the first i + 1 holes is the same in the unique winning position with s stones as in the winning position with s + q stones. The winning positions with s + 1 and s + q + 1 stones, respectively, are obtained from the corresponding positions with one fewer stones by either adding stones to the same hole in both cases or by adding stones to two different holes, each having index larger than i + 1. In either event, the effect upon the first i + 1 holes is the same and $p_{j,s+1} = p_{j,s+q+1}$, for j = 1, 2,. . . ,i+1. We are able to conclude that $p_{i+1,s} = p_{i+1,s+q}$ for all s≥0. Therefore the sequence of i+1-tuples has period q.

14.3 Asymptotics

There is a winning position for every number of stones given an unlimited number of holes. However, as in Ayo, if there is a finite number of holes n, then not all winning positions are realizable. In particular, rows in figure 4 of length larger than n can not be realized.

Let s(n) denote the smallest number of stones which actually requires the nth hole to win. (Obviously any greater number of stones will require at least n holes.) We will derive the asymptotic formula $s(n) \sim n^2/\pi$ from several interesting observations arising from an examination of the sequences $p_{i,s}$ and $m_{i,s}$.

Lemma 3: Given the above notations, $p_{i,s} - p_{i-1,s} = (i-1)(m_{i,s} - m_{i-1,s}) + 2m_{i,s}$.

Proof: Equations 1 and 2.

Lemma 4 *The sequence* $(m_{i,s})_{i=1}^{\infty}$ *is non-increasing.*

Proof: From Lemma 3

$$p_{i,s} - p_{i-1,s} = (i+1)(m_{i,s} - m_{i-1,s}) + 2m_{i-1,s}.$$

Since $p_{i,s} - p_{i-1,s} \leq i$, we have $m_{i,s} - m_{i-1,s} \leq 0$ and $m_{i,s} \leq m_{i-1,s}$ as needed.

Theorem 5: As n increases, s(n) grows as $n^2/\pi + O(n)$.

Proof: Let s be fixed. Define f(M) to be the least i such that M = $m_{i,s}$. By Lemma 4, $m_{i,s}$ = M if and only if $i \in I_M$ *where* $I_M = \{f(M), f(M)+1, \cdots, f(M-1)-1\}$. By Lemma 3, $p_{i,s} - p_{i-1,s} = 2M$ *for* $i, i+1 \in I_M$. Thus, the sequence $S_M = (p_{i,s})_{i \in I_M}$ is a finite arithmetic sequence with common difference 2M.

Now,

$$p_{f(M),s} - p_{f(M)-1,s} = (f(M)-1)(M - m_{f(M)-1,s}) + 2M.$$

Since $p_{f(M)-1,s} \le f(M)-1$ *and* $M - m_{f(M)-1,s} \le -1$, we see that $p_{f(M),s}$, the leading term of S_M satisfies

$$0 \le p_{f(M),s} \le 2M.$$

A similar argument shows $p_{f(M-1)-1,s}$, the final term of S_M, satisfies

$$f(M-1) - 2M + 1 \le p_{f(M-1)-1,s} \le f(M-1) - 1.$$

To compute the total number of terms f(M-1) – f(M) in S_M we computer the difference of the leading and final terms, divide by the common difference and add one.

$$f(M-1) - f(M) = \left(p_{f(M-1)-1,s} - p_{f(M)}\right)/2M + 1$$
$$= f(M-1)/2M + k_1$$

where $|k_1| \le 3$. Hence

$$f(M) = \frac{2M-1}{2M} f(M-1) + k_2$$

where $|k_2| \le 3$, or explicitly

$$f(M) = \frac{1x3x5x\cdots(2M-1)}{2x4x6x\cdots x2M} n + k_3 M$$
$$= \frac{\frac{1}{2}\frac{3}{2}\cdots\frac{2M-2}{M} n + k_3 M}{M!}$$
$$= \frac{\Gamma\left(M+\frac{1}{2}\right)n}{M!\sqrt{\pi}} + k_3 M$$

where $|k_3| \le 3$, since $\Gamma(1/2) = \sqrt{\pi}$ taking n+1 = M(0).

We must compute $s(n) = \sum_{i=1}^{n} p_{i,s}$. Thus, we are led to the sum of each sequence I_M. The number of terms has been already computed to be f(M-1)/2M + k_1, where $|k_1| \le 3$. Furthermore, the average term is

$$\left(p_{f(M-1)-1,s} + p_{f(M),s}\right)/2 = f(M-1)/2 + k_4 M$$

where $|k_4| \le 1$. Multiplying, we find the sum of I_M to be $f(M-1)^2/4M +)(f(M-1))$. We are thus led to calculate

$$s(n) \sim \sum_{M=1}^{\infty} \frac{\Gamma\left(M+\frac{1}{2}\right)^2 n^2}{4\pi M!(M-1)!} = \frac{n^2}{4\pi}\,{}_2F_1\left(\tfrac{1}{2},\tfrac{1}{2};2;1\right)$$

The result follows from Gauss's summation formula [5], since the hyper-geometric function ${}_2F_1\left(\frac{1}{2},\frac{1}{2};2;1\right)$ is equal to 4.

14.4 References

[1] Duane M. Broline and Daniel E. Loeb, "The Combinatorics of Mancala-Type Games, Ayo, Tchoukaillon, and 1/π", UMAP J. 16(1), 1995, pp 21–36.

[2] A. O. Odeleye, Ayo: A popular Yoruba game, Oxford University Press, Ibadan, Nigeria (1977).

[3] Pierre Duchet and Pierre Audin (editors), Acts of MATh.en.JEANS congress, Ecole Polytechnique, April 23-26, 1993, Palaiseau, France.

[4] Andre Deledicq and Assia Popova, "wari et solo: le jeu de calculs africain," Collection Les Distracts 3, Cedic, Paris (1977).

[5] Carl Friedrich Gauss, Disquisitiones generales circa seriem in_nitani. . . , Comm. soc. reg. sci. Gott. rec. Vol. II (1813), reprinted in Werke 3 (1876) 123 - 162.

Chapter 15

Sequences in African Board Games

15 Introduction

Integer sequences occur frequently in African board games. For every state-space a sequence exists. One of the major benefits of future studies of the board games would be to identify the integer sequences and their properties. One obvious sequence is the Levine family of sequences. Levine's sequences [1] are found in some of the self-propagating states of African board games. We have discussed the patterns of such state spaces in previous chapters. In this section we provide the generating functions that can be used to produce the states. Before we show which ones they are, it is important to understand how Levine's sequences are obtained. We discovered these sequences without serious research. Hence it is our opinion that many more of such sequences related to African board games exist and are waiting for discovery. This chapter is provided as a resource to introduce the topic for further research.

15.1 Levine's Sequence

In the summer of 1997 Lionel Levine [2] submitted a new sequence to Sloane [1] to include in his table of sequences. The sequence is constructed via the array in this table.

Index of Sequence, i	Values of sequence
1	1 **1**
2	1 **2**
3	1 1 **2**
4	1 1 2 **3**
5	1 1 1 2 2 3 **4**
6	1 1 1 1 2 2 2 3 3 4 4 5 6 **7**
7	1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 7 7 7 8 8 9 9 10 11 12 13 **14**

Table 12: The array that produces Levine's sequence {1, 2, 2, 3, 4, 7, 14, . . .}

15.1.1 Generator Algorithm

In Table 1 if a row of the array contains the numbers

$$a_1 \quad a_2 \quad a_3 \quad \cdots \quad a_{k-1} \quad a_k$$

then the next row contains

$$a_k \ 1's, \qquad a_{k-1} \ 2's, \qquad a_{k-2} \ 3's, \quad \cdots$$

We obtain Levine's sequence by taking the last term in each row:
{1, 2, 2, 3, 4, 7, 14, 42, 213, 2837, 175450, 139759600, 6837625106787, 266437144916648607844, 508009471379488821444261986503540, …} (1)

The terms grow unexpectedly rapidly! The nth term L_n is

(i) the sum of the elements in row n - 2
(ii) the number of elements in row n - 1
(iii) the last element in row n
(iv) the number of 1's in row n + 1
…

Furthermore, if s(n, i) denotes the sum of the first i elements in row n, then we have

(v) $L_{n+2} = s(n, L_{n+1})$

(vi) $L_{n+3} = \sum_{i=1}^{L_{n+1}} s(n,i)$

(vii) $L_{n+4} = \sum_{i=1}^{L_{n+1}} \binom{s(\mathrm{n},\mathrm{i})+1}{2}$

The latter identity was found by Allan Wilks [3], who also found a more complicated formula for L_{n+5}, and used it to compute the last two terms shown in (1). No other terms are known!

An estimate of the rate of growth of the sequence was provided by Bjorn Poonen and Eric Rains [4]. They showed that

$$\log L_n \sim c\,\tau^n, \text{ where } \tau = \left(1+\sqrt{5}\right)/2$$

15.1.2 Summary of Proof

(a) Since generally $L_{n+2} \le L_{n+1} L_n$; $\log L_n$ is bounded above by a Fibanoci - like sequence.
(b) The sum of the (n + 1)st row is at most

$$\binom{\left[\frac{L_{n+2}}{L_n}\right]+1}{2} L_n$$

This implies

$$\frac{L_{n+3}}{2L_{n+2}} \ge \frac{L_{n+2}}{2L_{n+1}} \frac{L_{n+1}}{2L_n}$$

so that $\log\left(L_{n+1}/2L_n\right)$ is bounded by a Fibanoci - like sequence.

Colin Mallows [5] has determined numerically that a reasonably good approximation to L_n is given by the expression

$$\frac{1}{c_1} e^{c_2 \tau^n}$$

where $c_1 \approx 0.277, c_2 \approx 0.05427$.

15.2 Winning Positions in N-Hole Mankala

Playing a board game requires a logical mind, but also teaches a mind to be logical. Logical thinking is required at both the start, middle and endgame strategies. We have covered requirements for strategic logical moves using game defining matrices. End game strategies are considered in this section.

The game begins with n stones placed anywhere except in hole 0. A move consists in picking up the stones in some hole and placing one in each lower numbered hole. If the last stone falls in hole 0 then play continues, otherwise the game is lost. The objective is to get all the stones into hole 0.

The game is interesting because there is a unique winning position for any number of stones. These winning positions are shown below, and can be found by playing the game backwards.

n	Position
0	0
1	1
2	20
3	21
4	310
5	311
6	4200
7	4201
8	4220
9	4221
10	53110
11	53111
12	642000
13	642001
..	. . .

Table 13: The unique winning position for n stones in Tchoukaillon solitaire

The array can be more explicitly constructed by the rule that if the first 0 in a row (counting from the right) is in position i, then the next row is obtained by writing i in position i and subtracting 1 from all earlier positions. The sequence of successive values of i is {1, 2, 1, 3, 1, 4, 1, 2, 1, 5, 1, 6, 1, 2, …}.

Let t(k) denote the position where k occurs for the first time in this sequence. The values of t(1), t(2), t(3)…, are: {1, 2, 4, 6, 10, 12, 18, 22, 30, 34, 42,…}. This sequence has some very nice properties. It has been investigated by (in addition to the references mentioned above) David [6], Erd}os and Jabotinsky [7] and Smarandache [8].

(i) t(n) can be obtained by starting with n and successively rounding up to the next multiple of n _ 1, n _ 2,…, 2, 1. E.g. if n = 10, we obtain

$$10 \to \quad 18 \to \quad 24 \to \quad 28 \to \quad 30 \to \quad 30 \to \quad 32 \to \quad 33 \to \quad 34 \to \quad 34 \to ;$$

so t(10) = 34.

(ii) The sequence can be obtained by a sieving process: write 1, 2,… in a column. To get the second column, cross off 1 and every second number. To get the third column, cross off the first and every third number. Then cross off the first and every fourth number, and so on (Table 3). The top number in column n is t(n). Comparison of Tables 2 and 3 shows that connection with the solitaire game.

1	~~1~~							
2	2	~~2~~						
3	~~3~~							
4	4	4	~~4~~					
5	~~5~~							
6	6	6	6	~~6~~				
7	~~7~~							
8	8	~~8~~						
9	~~9~~							
10	10	10	10	10	~~10~~			
11	~~11~~							
12	12	12	12	12	12	~~12~~		
13	~~13~~							
14	14	~~14~~						
15	~~15~~							
16	16	16	~~16~~					
17	~~17~~							
18	18	18	18	18	18	18	~~18~~	
19	~~19~~							
20	20	~~20~~						

Table 14: A sieve to generate the sequence t(1); t(2); : : : = {1, 2, 4, 6, 10, 12, 18, …}.

At stage n, the first number and every nth are crossed off.

(iii) Finally, Broline and Loeb [9] (extending the work of the other authors mentioned) show that, for large n,

$$t(n) = \frac{n^2}{\pi} + O(n)$$

It is a pleasant surprise to see emerge from such a simple game.

15.3 References

[1] N. J. A. Sloane, "My favorite integer sequences", Information Sciences Research, AT&T Shannon Lab Florham Park, NJ 07932-0971 USA, njas@research.att.com
[2] L. Levine, personal communication (with Sloane), 1997
[3] A. R. Wilks, personal communication (with Sloane), 1997
[4] B. Poonen and E. M. Rains, personal communication (with Sloane), 1997
[5] C. L. Mallows, personal communication (with Sloane), 1997.
[6] Y. David, On a sequence generated by a sieving process, Riveon Lematematika 11 (1957), 26 - 31.
[7] P. Erd}os and E. Jabotinsky, On a sequence of integers generated by a sieving process, Indag. Math. 20 (1958), 115 - 128.
[8] F. Smarandache, Only Problems, Not Solutions!, Xiquan. Publ., Phoenix and Chicago, 1993.
[9] D. M. Roline and D. E. Loeb, The combinatorics of Mancala-type games: Ayo, Tchoukaillon, and 1/π, J. Undergraduate Math. Application 16 (1995), 21 - 36.

Chapter 16

Game Rules

16 Segmenting the Games

There are at least 300 versions of African board games involving holes and sowing of seeds. They fall into three major groups which we distinguish by the number of rows the board has. The most popular types are two row boards, for example, Oware. Many of them have storage areas at the ends of the board in which captured seeds are kept. Many of them do not permit re-entrancy. In other words, once a seed is captured, it is not re-sown into the board again. Re-entrancy is however a feature of many other African board games.

The board games normally have a varying number of holes ranging from 5 to 36 per row. The two rows 6 holes long board is the most popular. They are to be found predominantly in West Africa and the Caribbean. The name Oware refers to this version mostly.

The most popular of the four-row types are Omweso (Uganda) and Bao (Zanzibar). They are predominant in East Africa. The three-row type including Gebeta is the least popular and is found mostly in Ethiopia and Eritrea, mostly in the boarder regions of the two countries.

The most elaborate board type has 4 rows and 36 holes per row. This version is found mostly in Southern Africa. Moruba in the Limpopo area of South Africa is an example.

16.1 Mancala

The name Mancala is a misnomer of the African board games as this name is hardly used by the natives. Once the traditional name is used however, it becomes easier to ascertain the rules for that specific type. Using the colonial generic name blurs the distinction between the rules.

In the board games popularly called Mankala the end storehouses are necessary to for playing the game. For example, when sowing the seeds, sowing into the store holes to capture seeds is allowed. A few of the board games are in software. Only one is known to have been produced to fit into a mobile phone – Bantumi.

Nokia has made Bantumi a feature of its mobile phones. There are other African board game software online. My favourite is the Bao software at [1]. The site provides a Macromedia Flash enabled game with typically Afro settings.

16.2 References

[1] http://www.baogame.com/game.html

Chapter 17

Mancala Rules

17 Mankala

Mankala is traditionally played in the Arabic speaking part of the world. The word Mancala is derived from the Arabic word Mankaleh meaning to move. Although Mancala is the favourite generic name for all Count and Capture games amongst academics, it is hardly used by African natives to which the games belong and from where they originate.

17.1 Objective

The objective of the game is to capture as many seeds as possible, usually more than your opponent.

17.1.1 Game State-Space at Start

The board consisting of two rows has six holes per row. The board may have a store hole at either ends. Three seeds are deposited in each hole on the board. Each player owns a row of six holes nearest to him and the store hole on his right. As well as being used as a storehouse, seeds are sown into the end holes forming an integral part of the game. There are many versions of the board game that do not have storehouses and we will identify them when they are encountered.

17.1.2 Starting

Each player takes it in turn to start. The player chooses a hole from their own territory, from which all the seeds are taken and sown around the board in an anti clockwise direction.

17.1.3 Sowing seeds

The seeds are sown in an anticlockwise direction. One seed is placed in each hole travelling around the board both on your and the opponent's side. Seeds are also placed in the player's own storehouse as he goes around the board but not in the opponent's storehouse. If the last seed sown falls in your storehouse you may play again. You may continue playing as long as your last seed continues to fall in your storehouse.

17.1.4 Capturing seeds

Seeds may be captured from either side. Seeds are captured when the sowing of seeds by either player falls in an empty hole on their side. If the opposite hole on the opponent's territory contains seeds they are captured along with the last seed sown. If there are no seeds in the opposite hole nothing is captured. If a player's last seed falls into an empty hole on the opponent's side and the opposite hole on their own side has seeds in it, no seeds are captured.

17.1.5 End of game

There are three end-of-game conditions. The game ends when

a) all the seeds have been captured or

b) when all the seeds are on one side of the board or

c) when one player has captured 25 seeds or more.

The winner is the player with the most seeds in their storehouse.

17.2 Common Features

There are attributes and features that are common to many African board games. The common features are:

- Playing in anticlockwise direction
- Holes arranged in rows
- Sowing of one seed per hole until the seeds taken are exhausted
- Taking of all seeds in a hole to sow
- Capturing of seeds starting from the last hole a seed is sown
- Each player owns a territory
- Starting of playing from your own territory

Chapter 18

Abapa Rules

18 Abapa

Abapa is a traditional variation of Oware that is played by adults and used for competitions. Abapa is played mostly among the Akan people of Ghana. The word means “the proper version”. This is a literal translation from the Twi language spoken by the Akan of Ghana. This version is known by various names in West Africa and the Caribbean. Some of the names are Ayoayo (Yoruba-Nigeria), Awale (Ivory Coast), Ouri (Cape Verde), Warri (Antigua, Barbados), Adji-Boto (Ewes-Ghana & Surinam), Awele (Ga's-Ghana & Ivory Coast).

18.1 Objective

The objective of the game is similar to that of Makala to capture as many seeds as possible. The first player to capture 25 seeds or more wins the game. A draw is possible in this game when each player captures 24 seeds. If the last seed sown falls in the opponent’s hole and it contains two or three seeds, they may be captured.

18.1.1 Game State-Space at Start

The board is made up of two rows of six holes each. Four seeds are placed in each hole on the board. Each row of six holes is the territory of the player sitting nearest to them. Two holes at either ends of the board are used as storehouses.

18.1.2 Board

The standard Oware board consists of 12 holes distributed equally into two rows of six holes. Two extra holes at either ends of the board are also provided in some boards and are used as storehouses. Boards are usually made of wood and the seeds come from specific plants. Each player has six holes on his side of the board in the case of the portable boards. The game is played with 48 seeds, with 4 seeds placed in each hole.

18.1.3 Strategy

The players decide who to start the game. Between friends this decision is simple as compromises are made. In tournaments, this can be done tossing a coin or by mutual agreement. After the first game, the winner starts. The player starting the game picks up all the seeds from any one of the 6 holes in front of him from his territory. The player sows the seeds in an anticlockwise direction, in a consecutive manner. The first seed is placed in the hole to the right of where they were taken from. The remaining seeds are placed in the holes directly following each other without skipping a hole.

18.1.4 Omitting a hole

If a hole has more than 11 seeds no more seeds may be sown into it. Under this condition it is possible to place a seed in each hole until the player arrives at the original hole that the seeds were taken from. It is skipped with a seed sown into the hole after it. The same is repeated on subsequent rounds. This rule distinguishes Abapa from many other versions of Oware.

18.1.5 Capturing seeds

The strategy for capturing is to play so that your last seed falls in an opponent's hole with one or two seeds already in it, so that after playing, the holes have two or three seeds. These two or three seeds may be captured. If your last seed sown results to a sequence of two or three seeds all of them may be captured.

18.1.6 Multiple Captures

Multiple captures are allowed if your last seeds are sown into an opponent's holes so that after sowing you have a sequence of 2 or 3 seeds on his side. As long as the sequence is not broken by the presence of either empty holes or holes with seeds other than two or three, multiple captures is allowed up to a maximum of five holes in sequence. This means multiple capture results to your taking a maximum of 15 seeds. If the sequence is longer than five holes, a situation that is possible in a board of six holes per row, the player forfeits everything, as this will leave the opponent with no seeds to play with. This limit of 5 holes is a major distinction of this version.

18.1.7 Compulsory moves

Seeding or feeding of your opponent's holes with seeds when they are all empty is recommended in Abapa. If the situation arises where one player has no seeds (zero seeds in all the holes) to play with the other player must provide some seeds to the opponent if possible. This means the next move should be that which results to depositing seeds in the opponent's holes so that he can play. A move that does not result to this is not allowed. Having no seeds to play with and yet not capturing enough seeds to win the game is very possible. One of such situations is when one player has eight seeds in each hole on his side. In this situation he has not captured any seeds. Although this is the extreme case, there are other cases where one side of the board will have most of the seeds and the captured seeds are less than 25. Greater priority is placed on capturing seeds to the end of the game. Therefore manoeuvring seeds in such as to result to the opponent not having seeds to play with is discouraged.

18.2 End of game

There are two situations under which the game ends. The game ends when

a) one player has captured 25 seeds or more

b) the game has resulted to circular moves and both players decide that continuing will only lead to going round in circles in such a case each player keeps the seeds on their side

Chapter 19

Tampoduo Rules

19 Tampoduo

Tamoduo is another version of Oware a traditional Ghanaian board game played by children. The name is derived from the Twi language spoken amongst the Akans of Ghana and means "to collect a big bunch of seeds". It is descriptive of the nature of play of this version of the game. The same version is known along the West African Coast by various names. Among the Yoruba people of Nigeria it is also known as Ayo J'odu. In Somalia it is called Layli Goobaly. Although Layli Goobaly has a few extra rules the principal rules are the same.

19.1 Objective

The objective of the game is to capture as many seeds as possible from your opponent.

19.1.1 Game State-Space at Start

The board is made up of two rows of six holes and each player owns the row (territory) next to him. Four seeds are deposited in each hole on the board. When a board has extra end holes they are used to store captured seeds. These storehouses are not sown into.

19.1.2 Starting

Each player takes turns to start. The player chooses a hole from their own territory, from which all the seeds are taken and sown one seed at a time.

19.1.3 Sowing seeds

The seeds are sown in an anticlockwise direction placing one seed in each hole as one traverses the board. Tampoduo permits relay sowing. If the last seed drops in a hole with seeds in it, all the seeds are taken and then sown until such a time the last seed falls in a hole that is empty on either side of the board. If before a player wins the game and at a point in the game one player does not have any seeds, the other player must sow seeds from a hole that will result to the opponent having seeds to continue playing the game. If this ***"seeding"*** of the opponent is not possible then the game comes to an end. The remaining seeds go the player who has the seeds in their territory.

19.1.4 Capturing seeds

Seeds are captured when the sowing of seeds by either player falls in an empty hole on their side. If the opposing hole on the opponent's territory contains seeds they are captured as well. If there are no seeds in the opposite hole nothing is captured. If a player's last seed falls into an empty hole on the opponent's side and the opposing hole on your own side has seeds in it, no seeds are captured. In Tampoduo, there is no restriction as to the number of seeds that may be captured from a hole.

19.1.5 End of game

The game ends under the following three conditions:

a) when there are only two seeds left, one on each side of the board or

b) when there are too few seeds for any meaningful game to continue or

c) when one player has captured 25 seeds or more

Chapter 20

Nam-Nam Rules

20 Nam-Nam

Nam-Nam is a variant of Oware played by children of the Akans of Ghana. This Twi language word means “to roam” and is descriptive of the nature of play of this version.

This version of Oware is also called Jerin-Jerin by the Yoruba people of Nigeria. You will find the same version in the Caribbean, particularly in Antigua where it is call Round & Round.

20.1 Objective

The objective of the game is to capture the opponent’s territory. In other words, you may leave the opponent with no seeds to play. In fact this is the desirable outcome in other to win the game. This ultimate aim is usually achieved after at least several rounds. This strategy therefore teaches the players that it is appropriate and acceptable in war situations to capture the territories of your enemies and to capture all of their territories if the opportunity is available. Of course doing so should end a war. Does it?

20.1.1 Game State-Space at Start

The board is made up of two rows of six holes each. The two extra holes on each end of the board are used as storehouses. Four seeds are placed in each hole of the board. Each row of six holes becomes the territory of the player sitting nearest to them.

20.1.2 Starting

Each player takes turn to start. The player then chooses a hole from their own territory to start. All the seeds are taken and sown one at a time.

20.1.3 Sowing seeds

The seeds are sown in an anticlockwise direction placing one seed in each hole as the player traverses the board. Recursive or relay sowing is permitted. That is if the last seed drops in a hole with seeds in it, all the seeds are taken and then sown until a last seed falls in an empty hole. If at any point in the game a player does not have any seeds to play with, the other player must feed or seed the opponent’s territory with seeds from a hole that would provide the opponent with seeds to continue playing the game. If this is not possible then the game comes to an end, with the remaining seeds going to the player who has the seeds on his territory.

20.1.4 Capturing seeds

The number of seeds to capture per game varies in this variant of Oware. Seeds are captured when the sowing of seeds by either player creates four seeds on ones territory. This distinguishes Nam-Nam from other variants of Oware in which the number of seeds that can be captured is 2 or 3. One can only capture seeds on ones own territory, except in the course of your sowing seeds your last seed sown falls in your opponent's territory and that hole has four seeds. Captured seeds are deposited in ones storehouse.

20.1.5 End of round

When there are eight seeds left in play, since it is impossible to play for the last four seeds, the player who captures the penultimate set of four seeds, also gains the last set of four seeds on the board. This ends the round. The winner of the first round is the one who has captured the most seeds.

20.1.6 Second round

On completion of the first round each player deposits the seeds they have captured back in the holes in their territory four seeds in each hole. If each is able to fill the same number of holes then the result is a draw. However if one player is able to fill more holes than what they started with (they have more than 24 seeds captured) then those holes filled on the opponents side now become part of their territory in the next round. This represents expansion of territory. This is a unique Oware game encountered in which a player may expand his territory. This rule is unique to Nam-Nam and distinguishes it. The winner of each round is the one with the most territory. This capturing of territory can go on for quite a while as lost territory can always be reclaimed in subsequent rounds. In other words, the game simulates a battle and war situation. You may win some battles, but you have to win the war to be declared the winner.

20.1.7 End of game

The game ends when one a player has lost all his territory, meaning his territory is completely captured, all six holes.

Chapter 21

Awèlé Rules

21 Awèlé

Jean Retschizki, a professor of Educational Psychology at the University of Fribourg, Switzerland, visited the Ivory Coast and conducted a study of how different people played *Awèlé*. He offered this description in his paper [1]. In the paper he provides a bibliography citing published works about the game as observed by others over the years. Here is Retschizki's description of how to play the game:

Figure 22: An A

21.1 Objective

The objective is to capture seeds from an opponent’s territory until it is no more possible to capture or the holes are all empty.

21.1.1 Game State-Space at Start

The game of Awèlé is played by two players on a board with 2 rows of 6 holes. Each player has his own territory - the row on his side. At the beginning of the game, each hole must hold 4 seeds. Each player takes a turn. A move is made by taking all the seeds in a chosen hole of one's row and dropping them one by one, anticlockwise.

21.1.2 Capturing

The aim of the player is to capture seeds in his opponent's holes. A capture happens when the last seed dropped falls into a hole on the opponent's side that already holds one or two seeds, so that after the move, the hole now holds 2 or 3 seeds which are then captured. A player may capture several sets of 2 or 3 seeds, provided that all the sets are consecutive and on the opponent's side.

21.1.3 End of Game

The game ends when one of the following cases occurs:

a) There are so few seeds remaining on the board that it is not possible to capture any more

b) One of the players has no hole containing enough seeds to reach his opponent's side and his opponent's holes are all empty

c) The players decide by mutual agreement to stop playing and share the remaining seeds according to their analysis of the situation. This is the most common way of ending games between good players.

21.2 Reference

[1] Jean Retschizki, "Strategies of Expert Awèlé Players", *International Colloquium: Board Games in Academia III*, Florence, Italy, 1999, pages 84-94.

Chapter 22

Adi Rules

22 Adi

The board game Adi originates in Nigeria. The wooden board has two rows and six holes per row as shown in this figure below. The rules given here are credited to Felix Kpodo Amenu a student of Ohio Wesleyan University in 1955. There are sections of the rule that need to be verified as they differ significantly from what appears to be an acceptable set of rules. Those sections will be highlighted.

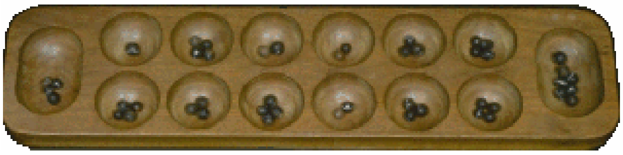


Figure 23: Adi Board [1]

At the age of about six years old Felix learned the game of Adi (*Ah'- dee*) from an old woman who herself learned it when she was young from another old woman.

22.1 Objectives

The objective of the game is to buy up all of the opponent's holes. The opponent can win back holes he has sold if at the end of a round he has more than enough seeds in his storehouse to fill the holes he still owns. He can win back as many holes as he can fill with groups of four seeds from his storehouse after he has filled all the holes he still owns.

22.1.1 Game State-Space at Start

The board has two rows of six holes each and two storehouses at either end of the board. Each of the two players owns one row (territory) nearest to him. At the start of the game, each hole has 4 seeds deposited and none in the storehouses.

A player takes seeds from any hole on his side, and sows them one to consecutive holes in an anticlockwise direction as far as the seeds can go. Relay sowing is permitted. If the last hole has more than one seed, he takes the contents of the hole in which the last seed falls, and sows them one to a hole, and continues the process until the last seed in hand falls into an empty hole, which ends his turn. This section is slightly modified compared to what was given in [2] as it appears there is an error in that documentation.

22.1.2 Capturing

Provided the last seed drops in a hole so that it contains 4 seeds, on either side of the board, the player captures them into his storehouse. During the game each player captures all 4's that appear on his side.

When only 8 seeds remain on the board, the player that captures the next 4 seeds takes all eight. This ends the first round.

Seeding of the opponent's holes is permitted. During the game, if the territory of a player is out of seeds, the other must play and feed seeds into them.

Re-entrancy is also allowed. For the next round, each player refills his holes from his own storehouse. The player who has a surplus fills empty holes. The rented holes temporarily belong to the winner. The loser may win them back for the next game.

As the players proceed, four seeds will accumulate in some of the holes again. Each player, even during the opponent's turn, quickly takes to his storehouse the groups of four seeds that appear on his side [this aspect of the rules is suspect and need to be verified].

If a player makes a hole of four when he drops his last seed, he takes that group even if it is on his opponent's side. Watch far this opportunity and try to win as many groups of four as you can. Also try to prevent groups of four from appearing on the opponent's side. When it is your turn, it is better to start in the hole that has more than four seeds rather than in holes with one, two or three seeds. This strategy enhances the chances of those one's, two's and threes of becoming four's. A player may sometimes start with one or two or three on his side if he thinks by doing so he prevents his opponent from getting four on his side. When a total of 8 seeds are left on the board, the player who captures the first 4 of them also takes the remaining.

22.1.3 End of Game

There are several situations for end of game. The winner of the game is the player with more than enough seeds in his storehouse to fill his holes with four each. The loser will not have enough seeds to fill all his holes with fours. The winner must fill those empty holes for the loser, and he thus buys the empty holes from his opponent by filling them with his own seeds. The winner may choose any of the holes from his opponent's side as the ones he is buying. It is advantageous to buy the holes in the middle of the opponent's side first, and work toward the outer ends. The one who loses the first game starts the next one. The players proceed as in the first round, except that now the groups of four that appear in the bought holes belong to the buyer. The only time a player can win a group of four in a hole he has sold to the user, is when he has formed four when ending in that hole. Now the winning player also may start in the holes he has bought on the opponent's side.

22.2 References

[1] "ADI: A COUNT AND CAPTURE GAME", Copyright 1955 Cooperative Recreation Service, Inc. Delaware, Ohio

[2] http://www.ahs.uwaterloo.ca/~museum/countcap/pages/adi.html

Chapter 23

Bantumi Rules

23 Bantumi

From the name of the game, it appears to originate from the Bantu speaking region of Africa. The game is played on a board of two rows and six holes per row [1]. There is a version for Nokia phones and also a palm pilot version called Palmtumi [2]. It runs on Palm OS v3.5.

23.1 Objective

The objective of the game is to move your seeds from the holes from your territory into your storehouse. The following rules apply to the Bantumi game.

23.1.1 Game State-Space at Start

At the start of the game, each hole in the two territories has four seeds in them (a total of 48 seeds, 24 per player).

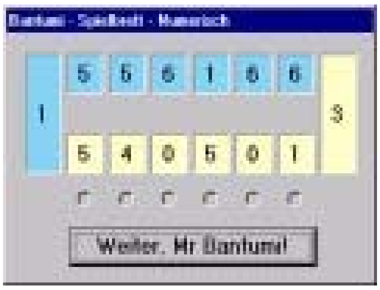


Figure 24: Bantumi Board

Figure 25: Palmtumi View

The seeds are sown in an anticlockwise direction. Choose a hole to play from and sow the seeds one to each hole following where the seeds were taken until they are all sown. Sowing into an opponent's storehouse is prohibited.

23.1.2 Capturing

If sowing results to a seed being deposited into your storehouse, your score increases by 1. If your last seed is dropped into your storehouse, continue sowing seeds. If the last seed in your hand falls in an empty hole belonging to you, that seed plus all seeds in the opposite hole are captured to your score. This is called "stealing" and is the most strategic part of the game, since you can "steal" many seeds from your opponent and add them to your score with this move.

23.1.3 End of Game

The game ends if the player to move has no legal moves (i.e. all his holes are empty. If there are still seeds in the opposite holes they are added to the opponent's score. The winner is the player with the highest number of stones in his storehouse.

23.2 References

[1] http://www.andybell.ch/bantumiapplet.php
[2] http://www.palmgamingworld.com/board/palmtumi.shtml

Chapter 24

Warri (Barbados) Rules

24 Warri

Warri is played outside Africa with some variations to the rules [1, 2]. The version described in this Chapter is found in the Barbados with 48 seeds on a rectangular board that has twelve holes arranged in two rows of six holes along the length of the board. Each player owns the territory next to him.

Figure 26: Warri Barbados

24.1 Objective

The objective of the game is to capture the majority of the seeds. All seeds have the same value, and the winner of the game is the player that captures more than half the number.

24.1.1 Game State-Space at Start

At the start of the game, 4 seeds are placed in each hole. Once it is decided who opens the game, the players take turns to play. In a move or "cut", the player lifts all the seeds from any one of the six holes that are on his/her side of the board, leaving that hole empty and redistributes or "sows" the seeds by placing one seed into each hole to the right of the chosen hole, without skipping any of the holes along the row. If the player still has seeds in hand when the redistribution reaches the rightmost hole on the player's side, he/she continues sowing seeds one by one into the holes on the opponent's territory of the board continuing the anticlockwise cycle around the board until the seeds in hand are exhausted. In a move therefore, the player repositions all of the seeds from one of his/her holes into the holes that follow along an anticlockwise path around the board.

24.1.2 Capturing

A player may only move from a hole on his/her side of the board. There is no limit to the number of seeds a player may accumulate in a hole. In fact it is a strategy to accumulate seeds in a hole or to "build a hole". This gives the player a long reach to seeds that he plans to capture. The trick is to know the right moment when you have the right number of seeds in the hole to break it! When a player chooses to move from a hole that contains more than eleven seeds, the distribution of the seeds will go a full lap around the board and commence a second lap but he must skip the hole from which the move was started. The hole from which a move was started should therefore always end up being an empty hole after that play.

Holes that contain counts of 1 or 2 seeds are said to be "vulnerable. When a vulnerable hole occurs on the opponent's side of the board, there is an opportunity for the player to make a capture. To take advantage of this opportunity, the player must use the move from a hole that contains precisely the right number of seeds, so that the last seed in the move comes to fall in the vulnerable hole. If such a move is possible, the player claims the contents of the vulnerable hole as well as the seed that was added in the capturing move. The total prize for the capture of a single vulnerable hole will therefore be two or three seeds. Captured seeds are removed from play and stored with the players other captures.

If there is a sequence of other vulnerable holes just before the captured hole, their contents are also automatically seized by the capturing move. As long as the sequence of vulnerable holes is unbroken or connected capture may take place. A player may only make a capture from a vulnerable hole on the opponent's side of the board.

24.1.3 Penalty

There is a penalty for leaving the opponent without seeds to move with. If the opponent's territory has no seeds remaining on it when it is his/her turn to play, the player forfeits all of the remaining seeds on the board to the opponent! Players should therefore always try to ensure that the opponent's territory has at least one seed with which to play. If a player has an opportunity to capture all of the seeds on the opponent's side as can happen in a multiple hole capture, he/she should be sure that the move will win more seeds than are lost in the penalty!

24.1.4 End of Game

The following situations end the game:

a) A player has captured most of the seeds on the board. The objective is to capture more than 24 of them. In fact, the game ends as soon as one of the players has captured 25 seeds

b) In the case where there are three or four endlessly circulating seeds (never ending moves), players agree to stop play and count the seeds they have captured. The player with the majority of the seeds wins the game.

c) A draw is declared if both players capture 24 seeds. In a tournament, the first player to win six games is the champion.

24.2 References

[1] http://barbadosphotogallery.com/warri/index.htm
[2] http://www.gamecabinet.com/rules/

Chapter 25

Owari Rules

25 Owari

Owari is very similar to Warri and Awale as the names suggest. The variant described here is the one played by the old men in market places in Lome, Ghana, and other parts of West Africa. It is rumoured that it used to be played by kings to prove their strategic skills on accession to the throne in Ghana [1]. A Palm Pilot version of the game is available at [2].

Figure 27: Owari Game Board

25.1 Game State-Space at Start

The board has two rows of six holes each. Each player owns a territory, one of the rows.

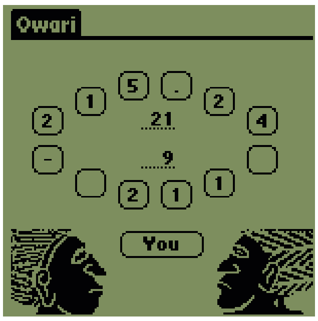


Figure 28: Owari – Palm Pilot View

Home: http://www.wizzy.com/owari/

25.1.1 Game State-Space at Start

At the start of the game, each hole in the board has four seeds in it for both players. Choose any hole in your territory to play from. Pick all the seeds in that hole and sow them one by one anticlockwise around the board and examine the hole where the last seed falls.

25.1.2 Capturing

Depending on the number now in this hole, there are different courses of action. If the last hole seeds other than 2 or 3, no action is taken and it is the other players turn. If the last hole is one of your opponent's and the number of seeds is 2 or 3, you capture those seeds. Examine the next set of holes and if they satisfy the same criteria, you collect those also.

If you play so many seeds that they wrap around the board (more than 11) you must skip the hole from which you picked up seeds. Do not sow into it.

You must provide for your opponent to be able to play by ensuring that the territory has seeds left. Rules are hazy regarding the case where your capture would remove all beads from her side.

25.1.3 End of Game

When you cannot feed a seeds, she collects all the beads left in the game, and the game is over.

25.2 Reference

[1] http://www.wizzy.com/owari/#Rules

[2] ftp://ftp.wizzy.com/pub/wizzy/palm/owari-1.2.zip

Chapter 26

Bao (Malawi) Rules

26 Bao (Malawi)

Bao is played predominantly in East Africa. The Malawi version is described in this Chapter. The description introduces how to sow the seeds, capture and win games. The intermediate and advanced games use the same basic strategy. Exceptions to the rules are provided. The exceptions create interest and balance. The terminologies used here have been given previously in Chapters 1 and 2.

The Kuu is set up to provide a hard-to-use opportunity for attack while being a major target. The introductory phase usually consists of attempts to sow your own Kuu to keep it safe, while stopping the opponent from doing the same.

The advanced set-up gives further opportunity to achieve the same ends, but it is necessary to be aware of opponents who exploit naive player's formations to wipe the opponents out in the first few turns.

Three sets of related rules are described in this Chapter including the basic, intermediate and advanced game rules. Bao as you will find from the descriptions below is a very smart game with rules that are very different from most other board games described in this book.

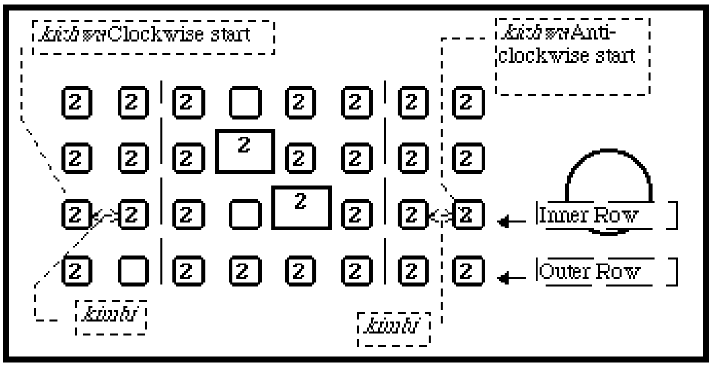


Figure 1 Bao Board with opening position for basic game

Italics denote Swahili words. Double quotation marks (") denote translated Swahili terms. Single marks (') denote terminology coined by researchers.

26.1 Rules of the Basic Game

The word bao refers to the board and contains four rows of eight holes. The opening position is shown in Figure 1. Sixty-four seeds are used, with two per hole.

The player to start the game is chosen by lot. In subsequent games the winner of the previous game plays first.

Each player owns a territory (2 rows nearest to him) and sows seeds only around the holes in his territory. The objective is to leave the opponent with an empty inner row. Players take alternate moves.

Recursive or relay sowing is permitted and may consist of several captures. A move starts by the player picking the seeds in one hole and sowing them one by one into the subsequent holes, moving either in a clockwise or anticlockwise direction, as desired. Note this uniqueness of Bao that permits clockwise moves.

26.2 Moves

There are two types of moves in Bao, mtaji and takata and they lead to three possible outcomes during the initial sowing:

26.2.1 Mtaji

- The last seed falls in an occupied inner-row hole and seeds are captured from the opponent's inner-row hole immediately opposite. A move of the first kind is called *mtaji*.

26.2.2 Takata

- The last seed falls in an occupied hole in the outer row, or one in the inner row opposite which no seeds are available for capture. In this case all the seeds in that last hole (including the one just dropped there) are lifted and sown (relay sowing) in the same direction.

- The last seed falls in an empty hole and the move ends and play passes to the opponent.

26.2.3 The Use of Kichwa

Captured seeds are sown along the player's inner row starting at the end-hole (kichwa) in the same direction as the previous sowing. Thus if the move leading up to the capture was anticlockwise the captured seeds will be sown starting at the anti-clockwise kichwa (Figure 1).

However, if the capture is from the first, second, seventh or eighth hole of the inner row (called *kimbi*) the captured seeds are sown from the nearest end-hole. This may mean that the direction of sowing is reversed. No change of direction is made when making a 'relay' (not a capture) from a kimbi hole.

During a mtaji, every subsequent relay sowing may occasion a capture if the circumstances described above recur.

A singleton cannot be lifted and sown on. A player left only with singletons to move has thus lost the game.

Fifteen is the highest number of seeds that can be used to begin a mtaji move. Any higher number of seeds can only be used for a takata move or, when a previous sowing ends there, for a subsequent relay of the same move, whether takata or mtaji.

Each move can be defined as mtaji or takata before it is played. If no initial lift endangers any of the opponent's seeds then takata is the only possibility. At no point subsequently during the same takata will a capture occur. Whether there are seeds in the opposite hole or not, moves ending in occupied holes cause a further relay.

A takata move is allowed only if no mtaji is available. If several mtaji possibilities exist, any may be chosen.

No takata move may be made from an outer-row hole if an alternative takata move is possible from the inner row. However, a mtaji move may begin from either row subject to the limits on the number of seeds involved.

All seeds in the outer row are safe, captures only occur from inner-row holes.

26.3 End of Game

The game ends when three conditions are fulfilled:

- A player left with only singletons to move has lost the game.
- A player loses the game when his inner row has no seeds.
- If an end hole is the only inner hole with seeds and a takata move is played back - across the player outer row, that player loses the game. The rationale is that the act of taking all the seeds in that end hole having the only seeds in the inner row causes the inner row to be empty during the move (even if the lone hole contains more than 8 seeds and so would eventually have reached back round onto the inner row).

26.4 Rules of the Intermediate Game

Play starts from the state-space (position) shown in Figure 2. The two holes (the fourth from the right of each player's inner row) called *kuu* play a prominent role at the intermediate game stage. There are special rules attached the use of kuu at the intermediate game. They are cut square and larger. The remaining 40 pieces are kept in the storehouse (or on the ground, in the non-playing hand, the player's lap, etc.).

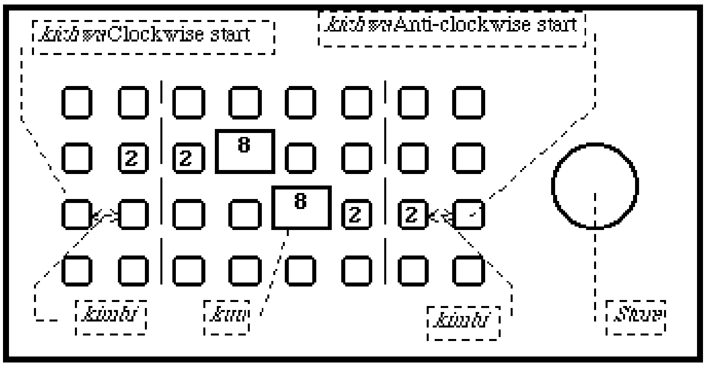


Figure 2: Opening Position for intermediate game

Play is in two stages. In the first stage those seeds not at first placed on the bao are introduced one by one. The second stage (where play continues as in the basic game) starts when all the seeds have been introduced into the board.

The first player adds one of the seeds put aside (called *nemo* in this context) to one of the holes other than the kuu. These are then sown in a takata move clockwise or anti-clockwise as the player chooses.

Play now continues, each player adding a nemo to an existing hole and capturing or making a takata move as appropriate. The players take turns to move.

Adding a nemo to an occupied hole with seeds in the hole opposite "captures" the seeds opposite, they are then taken and sown; sleeping, relaying or capturing as in the basic game.

It is not allowed to takata from the kuu hole during the first stage of play.

Singletons can be used to takata in the first phase since adding the nemo means they act as two seeds. However, this is allowed only if no inner-row holes (other than the kuu) contain more than one seed.

As noted in the basic game moves that result to captured seeds have precedence over takata ones. When there are seeds in any of the opponent's inner-row holes opposite occupied holes on the player's own inner row, the player must capture by placing the nemo in one such hole. This captures the seeds opposite and they are sown along the player's own inner row from whichever end the player chooses. If the capture is from a kimbi hole the captured pieces must be sown in from the nearest end-hole.

By capturing the seeds opposite an occupied hole the player protects the latter from obvious capture at the adversary's next turn. It is therefore usual to give precedence to protecting the kuu as this can be of vital importance. Should a player fail to protect the kuu, spectators will usually comment and the move may be retaken. This acts as an unofficial refereeing of the game of some sorts. In other board games, it is normal and usually advisable for spectators to keep silent as the game progresses, since by instructing or advising opponents conflicts could arise. In this special Bao rule, such conflicts do not exist as the situation is covered by a rule.

In the first phase a player may no add a nemo to directly take an opponent's kuu if the player's own kuu is threatened (i.e. there are one or more pieces in the opposite hole).

The kuu may be the only occupied hole on the player's inner row and there may be no seeds to capture opposite. In this circumstance a nemo is placed there and two seeds only are removed and sown to one side or the other (known as 'taxing' the kuu). There is one exception described below.

The seeds in one's kuu can only be lifted and sown during the first stage in the following two circumstances:

A move started with a capture and the last seed of a subsequent sowing is placed in the kuu. The player may now choose whether to "sleep" or lift and sow on ('activating' the kuu by a 'relay').

If it is not possible to capture and the kuu contains exactly eight seeds and the kuu is again the only occupied hole of the inner row a special rule applies. A nemo is placed there and all the nine seeds are lifted and sown.

If any lap of a takata move ends in the kuu, the player "sleeps"; the seeds cannot be lifted and sown.

Once a player's kuu has been emptied or compromised (but not just 'taxed') the hole loses all its privileges and restricting characteristics and becomes an ordinary hole like all the others. The kuu in any case have no special significance during the second stage of play except during the first takata as described below.

Just as in the basic game, a move that started as takata (i.e. without capture) cannot include captures in subsequent relays.

26.5 Extra Rules of the Advanced Game

In this case too, play starts from the state-space or position shown in Figure 2. The first player adds one of the seeds put aside (called *nemo* in this context) to the four seeds already laid out in holes other than the kuu. These five seeds may be distributed across the 16 holes on the player's side of the board, as desired.

The second player repeats this process. However the second player can also take the seeds from any one hole opposite those which contain seeds (the kuu or those containing two seeds). These captured seeds (if any) are added to the five to be distributed across the 16 holes on the second player's side of the board. Play then continues as outlined in the intermediate game.

26.6 General Points

If in the second stage of the game, a takata move leaves the opponent with only takata moves and the player with only one target for mtaji moves; then special restrictions apply to the next two moves:

The other may not now takata these threatened seeds to start the next move (unless these are the only seeds available to takata). Nor may they be lifted for a subsequent relay of the same takata move. If the last seed of any sowing lands there, the player 'sleeps'.

The initial player must then make the original threatened capture, whatever the consequences. If there is more than one source for the attack the player may choose between them.

Each player embarks on the second stage of the game immediately after exhausting the stock of nemo, by carrying out a move according to the basic rules.

If by the second stage of the game a player's kuu has not yet been moved it must be moved on the player's first takata move. The consequence of this move (and others with more than 16 seeds) is hard to calculate and of critical importance. The player can therefore try out the move, first in one direction and then in the other, reverting to the first if that seems best.

Except when defending the kuu or making the first takata move with the kuu in the seconds stage of the game, no re-taking of moves is permitted. Once touched (except for the clear purpose of counting), seeds must be lifted and sown.

In very exceptional circumstances, a player's takata move may relay around the bao many times and never come to an end. If this circumstance is apparently happening the player simply announces that it appears the move will not end naturally and "sleeps" at the end of the current relay.

26.7 Comments

The second player should always place the last nemo. Due to mis-play, it may happen that the last seed is taken from the store by the first player. In this case first player donates a seed from any back row hole containing more than one seed.

The restriction on re-taking of moves is often overlooked when playing with younger less experienced people or with good friends playing just to "push the time along", and typically in smaller villages or towns.

Any spectator wishing to point out a more advantageous move may do so without prejudice to the other player's game, thanks to the restriction against re-taking moves. The merits and demerits of the proposed alternative can be discussed while the seeds are still in their unmoved position, but the player must always make the original move.

Counting of one's own seeds is permitted. Counting the adversary's seeds is common but they may not be moved to do so. Counting is often done in such a manner as not to reveal to the opponent the number of seeds being counted.

Speed of decision is not at any great premium, especially once the game has reached some degree of complexity. An average game may last ten to fifteen minutes, but when two good players (or two incompetents) are matched it may last much longer.

A set is usually decided by the best of three games. When the winner of a set is known, the loser withdraws and a new player (normally chosen by order of arrival) takes over. If this new player beats a previous winner in their first game, the latter retires allowing yet another to play, otherwise they decide by best of three as usual. Great game!

26.8 References

[1] Pete Duckworth, Mark Chikoko and Adrian Brooks
[2] Philip Townshend,.**Bao (Mankala): The Swahili Ethic In African Idiom**

Chapter 27

Warri (Java) Software

(Visa Korkiakoski and Janne Pänkälä [1])

27 Introduction

There are several software and source codes on the Internet that implement versions of African board games. Many of them are based on Java. This Chapter is one of them and is included with the kind permission of Visa Korkiakoski and Janne Pänkälä [1]. Wherever the authors used the word "cup" we have changed it to hole. The software is easy to use and to compile. We have successfully compiled it under three Window versions (Windows 98, 2000 and XP) without having to do any debugging at all. The compilations used Java 1.4.2 under BlueJ development environment.

This chapter is provided to lead the student to understand what is required when implementing any of the rules described in the previous chapters. It highlights the sections needed and how the rules are used. It also provides the basis for extending the code to other board games as described in this book. The student will through reading and compiling the code for herself gain an invaluable insight into game design.

27.1 Warri Software

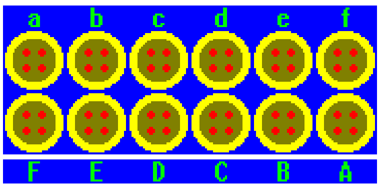


Figure 29: Board State-Space at Start of Game

The purpose of the game like all the ones described is to sow the seeds by taking all seeds from one hole and sowing them onwards to holes next to it in an anticlockwise direction. If sowing ends up on the opponent side and it ends up in a hole which has 2 or 3 seeds after sowing all the seeds the player whose turn it is can capture all the seeds in that hole and all the seeds in previous holes in the opponent's side if they have 2 or 3 seeds. Specific rules of Warri may be obtained from [2] on the Internet.

The game will normally continue until the board is empty or an unresolved situation occurs.

27.2 AI Features

Several features of artificial intelligence (AI) were used in writing the software for the game and they include:

- Minimax search
- Alpha-Beta pruning
- Limited peaceful search (goes one depth more if we are not in peaceful situation)
- unresolved situation check (implemented with 2 AI players that play against each other if the situation is such that unresolved situation may occur)

27.3 Unimplemented Features and Fixes

The following features have not been implemented. The reader may decide to enhance the code by implementing them for herself.

- Player(s) could start thinking about their move while animation is happening.
- Different strategies for computer (win, equal, lose) which could be selected below the computers difficulty.
- Undo button to help mere humans to cheat against computers overpowering wisdom.
- Proper dynamically changing strategy. Computer should aim for different goals depending on the progress in game (ie. how many seeds are still on the table)
- Ending test should be corrected. Now it observes the end of the game letting two third level computer players play against each other. In some cases more advanced computers get different ending result. Secondly, the ending test shares the seeds according to their current locations in the current state-space, but not to their locations in the ending situation.

27.4 Implementation Strategy

The game is implemented around 2 main classes (Figure 2), *Node State and Mancala Table.*

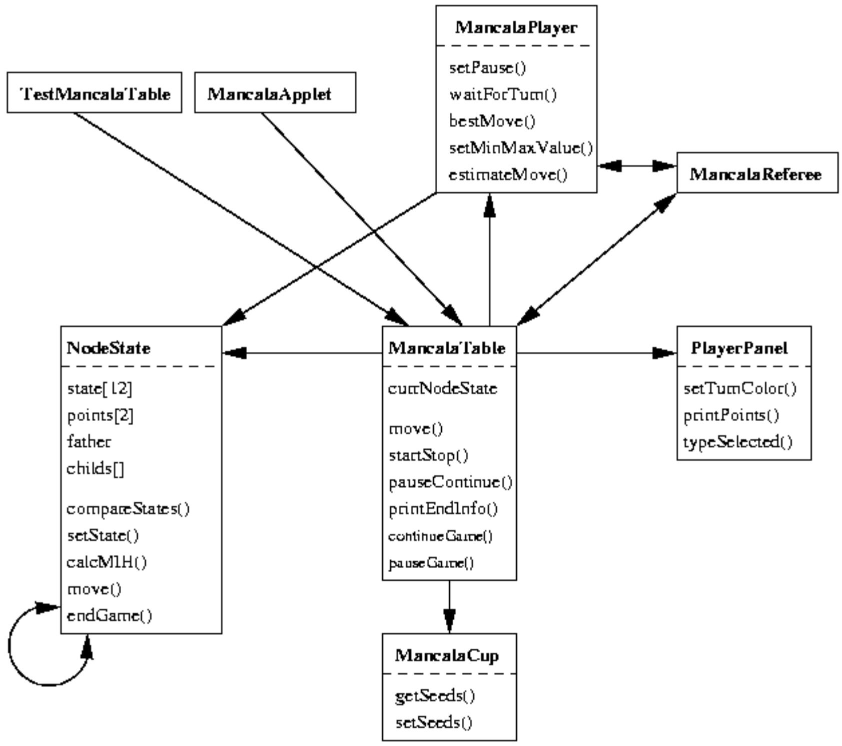


Figure 30: Relationships Between Java Classes Used to Implement Game

- **NodeState** is the area that knows the rules of the game and changes the game situation according to the rules. It also keeps the history of all the moves made and can be effectively used to "estimate" moves to come. The artificial intelligence module uses this very much.
- **MancalaTable** is responsible for refreshing the game board and calling the players and referee on the place and waking them up when needed. It therefore performs a supervisory role.

When the program is started we call MancalaTable contructor (either with or without some "debug" state). The ***MancalaTable*** will take care of the fact that everything else starts to work

All the *intelligence* is implemented in **MancalaPlayer** class. Specific information about the classes and class relations can be found in JavaDoc format from the authors' web site [1].

27.4.1 Implementation of the Game Theory

When the game starts the Player thread is woken up. If it' *intelligence* is 0 it means that it is a human player and we wait for event from game board. If we are dealing with computer player (IQ != 0) then it starts systematically evaluating moves with minmax.

Evaluation works in a way that the Player handles NodeState class methods and creates new states recursively (clone from the state that it is in at that moment) and then implements move to that state (which has previous state as father). When recursion reaches depth <= 0 and the state is *peaceful* it does evaluation of that leaf and tells its Father either the min or max value and returns. Depth first search and also Alpha-Beta pruning are implemented and if the path already is worse than what we have in some other path then the search in that path is interrupted and we move on to other possibilities.

Memory saving is implemented as in depth first search. When we have all the knowledge that we want from the child nodes we nuke them and this Java programming language (hopefully) frees the memory. Exception to this rule is the state where we started the search. It wont destroy its child for it needs to know which child has best max value and which move takes it there. So we nuke the starting state's child after we have ran bestMove() (which decides that next move) on it.

When the move is decided we inform the table which does the actual job of showing the move to us and setting us to new state.

Unresolved game testing is performed after each move if we have 10 or less seeds on the table and MIH difference is 2 or less. Testing works in a way that makes 3rd level computers to play against each other (this is very fast). Computers play against each other until they notice that something has been eaten (which means that we do not have a final situation at least yet) or we find identical situations from game history.

27.5 Results

Since we were (are) no experts in mancala strategy we resulted to huge amount of empirical testing to determine which estimation function would give the best solution. In these tests we put 2 computer players of various IQ levels to play against each other. Test results are available at [1] and reproduced in Appendix I to this chapter.

These empiric tests show that using MIH to estimate the value of game situation in far away leaves is not of much use. However it does speed up the search much since it makes Alpha-Beta pruning actually do some serious work by setting more variations in m{in, ax} values. This was noticeable in test case10.

Even if the results seems to be somewhat weird it can be noted that matrices are generally diagonal so that negative in lower left corner and positive in right top corner which means that smarter computer wins. This is especially noticeable in case10.

Also we noted that using MIH value to estimate next move where several equal maximums are available gives very noticeable results in favour of using MIH in bestmove(). The player that uses MIH value in addition to minimax estimation is generally victorious as shown in cases 6 and 7.

We also tested simple dynamic strategy in 8 and 9 which would select the bestmove differently depending on the amount of seeds on the board but this did not give results we hoped for since it showed it has no noticeable impact on the results. Perhaps more complex dynamic estimation would have been needed. However with our present Warri knowledge creating such would have taken the best part of this year.

27.6 Conclusions

According to the tests we performed we ended up with the solution obtained in the last test. Most of the stuff we tested while trying to improve the AI is kept in the code is spite of the fact we did not find any straight proof of improvements given by them. Since they cause no harm and may be interesting for human players to find, we considered it wise to preserve the small additions.

Even when computer does seems to produce some weird solutions - in cases where it plays against a computer of same level or computer of 1 level lower starts and defeats the smarter one - it does defeat the human player somewhat easily (at least for us ☺) and therefore we think it is adequate for the purpose of tormenting some poor fellows brain. However, it has been proven in the presence of witnesses that the fourth level computer can be beaten using human brains.

27.7 References

[1] http://www.hut.fi/~vkorkiak/mancala/doc/doc.html
[2] David Chamberlin, Chapter 1: The Rules of the Game, unknown, unknown http://scs.student.virginia.edu/~games/traditional/warri/chapter1.html.

Appendix I: Testing

Empirical Test Results for Mancala/Warri

Starting player IQ level is in vertical column on the left. Other players level is in the horizontal row at the top.

Results are differences in the end points: (Other - Starting) player points. For example in the table below when starting player is 1st level computer and it is playing against 4th level computer the result is 40 points in favor of the "wiser" player. And vice versa 4th level starting player wins 1st level "vegetable" player with 48 points.

A. Players are equal

1. Test where both players use MIH-knowledge in both leafs of the minimax estimation and next move which is found with bestMove()

0	1	2	3	4	5	.
1	36	-18	-16	-48	-16	.
2	36	40	22	-16	-22	.
3	30	8	4	-12	-6	.
4	40	30	8	-4	0	.
5	18	6	18	16	-8	.

2. Test where neither player uses MIH-knowledge in minimax leafs but uses it on bestMove()

	1	2	3	4	5	.
1	6	-14	-18	-20	-16	.
2	22	-8	-20	-20	-8	.
3	38	30	-6	-16	-2	.
4	6	20	18	-6	-8	.
5	6	18	6	-8	16	.

3. Test where neither player uses MIH-knowledge anywhere.

	1	2	3	4	5	.
1	2	-12	-22	-14	-22	.
2	4	10	-16	-14	-26	.
3	16	24	-14	10	-16	.
4	28	8	2	-6	-8	.
5	6	22	4	20	4	.

B. The other player does not calculate MIH in leafs

4. Test where starting player plays without MIH-knowledge in minimax estimation, but uses it to next move in bestMove(). Other player uses MIH also in minimax estimation.

	1	2	3	4	5	.

1	6	-22	-20	-20	-16	.
2	24	-12	-32	-20	-6	.
3	44	10	-8	2	0	.
4	22	16	8	16	-20	.
5	40	-20	22	-2	-12	.

5. Vice versa from case above. Other player does not test MIH in leafs but starting player does.

	1	2	3	4	5	.
1	36	-14	-22	-20	-16	.
2	44	-14	-24	-14	-22	.
3	36	-2	2	-14	-22	.
4	42	26	22	-38	-8	.
5	20	16	18	8	-18	.

C. The other player does not calculate MIH anywhere

6. Test where starting player does not use MIH-knoledge nowhere. Other player uses it both in bestMove() and leafs.

	1	2	3	4	5	.
1	-24	-40	-36	-24	-32	.
2	20	10	-4	14	0	.
3	26	32	10	4	-14	.
4	36	34	32	26	20	.
5	24	30	36	32	18	.

7. Vice versa from case above. Other player does not test MIH in anywhere but starting player does.

	1	2	3	4	5	.
1	18	-14	-26	-40	-8	.
2	12	-6	-36	-12	-12	.
3	14	-8	-4	-24	-8	.
4	18	-2	-2	-14	-26	.
5	24	8	20	-10	-8	.

D. Strategy for equal minimax values varies

8. Test for the old strategy.
If no differences are observed in possible moves, move the rightmost hole.

	1	2	3	4	5	6	.
1	36	-18	-16	-48	-22	-48	.
2	36	40	20	-16	-22	-6	.
3	32	8	2	-22	-6	-8	.
4	40	30	4	-8	0	-4	.
5	22	6	14	16	-14	-12	.
6	20	28	20	-2	4	16	.

9. Test for the new strategy.
If no differences are observed in possible moves, move the rightmost hole if there are more than 10 seeds in the table. Otherwise move the leftmost hole.

	1	*2*	*3*	*4*	*5*	*6*	*.*
1	*36*	*-18*	*-16*	*-48*	*-16*	*-44*	*.*
2	*36*	*40*	*22*	*-16*	*-22*	*-6*	*.*

3	*30*	*8*	*4*	*-12*	*-6*	*-8*	*.*
4	*40*	*30*	*8*	*-4*	*0*	*2*	*.*
5	*18*	*6*	*18*	*16*	*-8*	*-16*	*.*
6	*22*	*34*	*20*	*-2*	*4*	*16*	*.*

10. Extended test for the current (the same as previous) strategy.
Traditional test where both plays with MIH knowledge (checked in leafs to estimate minimax and also in bestMove() should such case occur that more than 1 child has same bestmax.).

	1	2	3	4	5	6	7	8	9	.
1	36	-18	-16	-48	-16	-44	-20	-36	-38	.
2	36	40	22	-16	-22	-6	-2	-22	-26	.
3	30	8	4	-12	-6	-8	-20	-30	-6	.
4	40	30	8	-4	0	2	-8	-28	-16	.
5	18	6	18	16	-8	-16	-4	-14	-4	.
6	22	34	20	-2	4	16	-6	-8	-4	.
7	36	34	28	0	18	38	-2	-4	-24	.
8	36	36	12	26	44	8	-6	4	18	.
9	36	32	12	14	22	14	8	16	2	.