

Generalized Bawo: A Finite-State Model and an EXPTIME Upper Bound

Isaac Kaimfa

September 2026

Abstract

Four-row mancala games, including Malawian Bawo and the related East African Bao, have received limited attention in computational complexity theory despite their combinatorial richness. This paper introduces GENERALIZED-BAWO-WIN, a finite-state two-player game model inspired by Malawian Bawo. The model is parameterized by a half-board width n and a binary-encoded lap cap L , and incorporates relay sowing, marker-pit capture, a formalized capture-relay rule (abstracting *kichwa*), mandatory capture, a singleton-start prohibition, and an explicit lap-cap truncation rule that guarantees move termination. We prove that GENERALIZED-BAWO-WIN belongs to EXPTIME under these semantics. The proof proceeds by establishing that the game's state space has size at most $2^{\text{poly}(|I|)}$, that legal successors can be generated in $2^{\text{poly}(|I|)}$ time, and that a WIN/LOSS/DRAW retrograde attractor can be computed over the resulting finite game graph. We emphasize that this is an upper bound: no hardness result is established. We further emphasize that the theorem applies only to the formally defined generalized model and does not classify Bao as played in East Africa, whose rules differ materially.

1 Introduction

Mancala games constitute one of the oldest and most widely distributed families of combinatorial board games, with hundreds of variants documented across Africa, the Middle East, and South and Southeast Asia [1, 2]. Among these, the four-row variants played in eastern and southern Africa are notable for their strategic depth and the complexity of their sowing and capture mechanisms. Two prominent members of this family are Bao, played primarily along the Swahili coast and in Zanzibar [3, 4], and Bawo, played in Malawi, particularly among the Yao people [5].

Despite growing interest in the computational analysis of traditional board games, the formal complexity-theoretic classification of four-row mancala games remains largely open. Previous computational work on Bao has focused on state-space estimates and game-tree search [6, 7, 8], but has not established membership in standard complexity classes such as PSPACE or EXPTIME. The combinatorial structure of these games, involving relay sowing, conditional captures, mandatory-capture constraints, and the possibility of very long or even perpetual moves, presents challenges that standard two-player game frameworks do not immediately address.

This paper introduces a formal computational model called GENERALIZED-BAWO-WIN. The model is inspired by Malawian Bawo but is defined as a precise mathematical object, parameterized by a half-board width n and a lap cap L encoded in binary. The model incorporates relay sowing (R1), marker-pit capture (R2), a formalized capture-relay abstraction (R3), mandatory capture (R4), a singleton-start prohibition (R5), and lap-cap truncation (R6). Rule R6 is a formal modeling choice that ensures every individual move terminates in a bounded number of micro-sowing steps, which in turn guarantees that the game graph is finite.

The central contribution is:

Theorem 1. *GENERALIZED-BAWO-WIN under binary- L lap-cap truncation semantics belongs to EXPTIME.*

The proof proceeds by bounding the state space, bounding successor-generation time, constructing the explicit finite game graph, and computing a WIN/LOSS/DRAW retrograde attractor over it. We are careful to distinguish configurations from legal states and from reachable states, and to handle the possibility of draws arising from cyclic play.

We emphasize what this result does not establish. It does not prove EXPTIME-hardness, PSPACE-hardness, or completeness for any class. It does not classify historical Bawo or Bao. The EXPTIME membership is an upper bound; the exact complexity may be lower. We discuss these limitations in detail and present a list of open problems.

Scope statement. This paper studies GENERALIZED-BAWO-WIN under the formally specified rules R1 through R6 with lap-cap truncation. It does not prove any complexity bound for Bao as played in coastal East Africa or Zanzibar, whose rules differ materially.

2 Related Work

The scholarly study of mancala games spans ethnographic documentation, combinatorial analysis, computational search, and complexity theory. We review relevant work in each category, taking care to distinguish the type of contribution each makes.

2.1 Ethnographic and Rule Scholarship

De Voogt [3, 4] has provided extensive ethnographic documentation of Bao as played in coastal East Africa and Zanzibar, including detailed rule descriptions, regional variation catalogues, and strategic commentary. His work documents the existence of never-ending moves in Bao, in which a sowing sequence can cycle indefinitely under certain configurations [3, 9]. This phenomenon is significant for complexity analysis because it implies that a naive finite-state formulation of Bao may not terminate without additional rules or conventions. Malawian Bawo, while related, has distinct rules regarding capture, sowing direction, and the treatment of special pits [5].

2.2 Combinatorial and State-Space Analysis

Kaimfa [5] provides a state-space complexity estimate for Malawian Bawo, giving an upper bound on the configuration space. That work addresses the number of representable board configurations but does not establish membership in a computational complexity class. The distinction between configuration-space size and complexity-class membership is important: a large state space does not by itself imply hardness, nor does a bounded state space by itself imply tractability.

2.3 Computational Search

Donkers and Uiterwijk [6] and Donkers, van den Herik, and Uiterwijk [7, 8] have investigated Bao from a computational game-search perspective, implementing Bao engines and studying search-tree properties. This work addresses the practical question of how to play Bao well computationally but does not establish complexity-class membership. The game-tree complexity and state-space complexity estimates in these works are empirical or combinatorial, not complexity-theoretic classifications in the sense of P, PSPACE, or EXPTIME.

2.4 Complexity Theory for Combinatorial Games

The general theory of two-player games provides the framework for this paper. It is well known that determining the winner of generalized versions of many board games is PSPACE-complete or EXPTIME-complete [10, 11]. For example, generalized chess is EXPTIME-complete [12], generalized checkers is EXPTIME-complete [13], and generalized Hex is PSPACE-complete [14]. The key structural features that drive these classifications include alternation depth, the polynomial-boundedness of game length, and the encoding of game parameters. To the best of our knowledge, no four-row mancala game has been rigorously classified in a standard complexity class prior to the present work.

3 Formal Generalized Bawo Model

3.1 Board

A generalized Bawo board is parameterized by a positive integer n , the half-board width. The board contains $4n$ pits arranged in four rows of n pits each. The pits are indexed as follows:

- A-back: pits $0, 1, \dots, n-1$
- A-front: pits $n, n+1, \dots, 2n-1$
- B-front: pits $2n, 2n+1, \dots, 3n-1$
- B-back: pits $3n, 3n+1, \dots, 4n-1$

Player A controls pits 0 through $2n-1$, and player B controls pits $2n$ through $4n-1$. Each player's front row faces the opponent's front row, with pit $n+j$ (A's front) aligned with pit $3n-1-j$ (B's front) for $0 \leq j < n$.

3.2 State

Definition 1 (State). A state of the generalized Bawo game is a tuple $s = (B, r_A, r_B, T)$, where: $B = (b_0, \dots, b_{4n-1})$ is the vector of nonnegative integer pit occupancies; r_A and r_B are nonnegative integer reserve counts for players A and B respectively; and $T \in \{A, B\}$ is the player to move.

The total number of seeds in play is $S = \sum_i b_i + r_A + r_B$. In the initial position this total is S_0 , a parameter of the game instance. Under the rules specified below, seeds may be discarded by the truncation rule R6, so S is nonincreasing over the course of play.

We distinguish three classes of configurations:

- C : the set of all binary-representable configurations (B, r_A, r_B, T) with nonnegative entries.
- S : the set of well-formed states satisfying $0 \leq \sum_i b_i + r_A + r_B \leq S_0$ and all pit/reserve constraints.
- S_{reach} : the set of states actually reachable from the initial position under the rules R1 through R6.

In general, $S_{\text{reach}} \subseteq S \subseteq C$, and all three sets may differ. The complexity proof operates on S (or equivalently an upper bound thereof); the reachable set S_{reach} is not characterized exactly.

3.3 Rules R1 through R6

R1: Relay Sowing. A legal move begins by selecting a starting pit on the moving player's side that contains at least two seeds. The player picks up all seeds from the starting pit and sows them one by one in a fixed cyclic order over that player's own two rows. There is no free direction choice: the cyclic order is determined by the board geometry. If the final seed of a sowing lap lands in an occupied pit (a pit that contained at least one seed before the final seed was dropped), the player picks up all seeds from that pit (including the just-dropped seed) and continues sowing. This relay process continues until the final seed of a lap lands in a previously empty pit, subject to the lap-cap constraint R6.

R2: Marker-Pit Capture. During relay sowing, when the moving player's sowing lands in an occupied pit on the player's own front row, the aligned opponent front-row pit may be captured according to a formal capture predicate. Specifically, if the aligned opponent front-row pit contains at least one seed, those seeds are removed from the opponent's pit and added to the moving player's reserve. The landing pit itself remains occupied and sowing may continue via R1.

R3: Kichwa Capture-Relay (Modeling Abstraction). In the generalized model, *kichwa* is represented as a landing-pit capture-relay associated with a designated front-row pit of the player to move. When relay sowing lands in the designated kichwa pit, a capture-relay sequence is triggered according to the formal rule specification.

Important qualification. This is a modeling abstraction. It does not constitute a complete ethnographic definition of kichwa as practiced in any particular regional variant of Bawo or Bao. Changing this abstraction defines a different formal game, potentially with different complexity properties.

R4: Mandatory Capture. If at least one legal starting pit produces a capture (via R2 or R3), then all non-capturing starting pits are illegal for the current move. The implementation must therefore perform capture detection over all candidate starting pits before accepting a move. This mandatory-capture lookahead is a significant computational step whose cost must be accounted for in the successor-generation analysis.

R5: No Singleton Starts. A pit containing exactly one seed is not a legal starting pit. Therefore, a position in which every pit on the moving player’s side contains at most one seed (or is empty) has no legal moves. Such a position is terminal under the game semantics defined below.

R6: Lap-Cap Truncation. Each atomic move has a maximum of L micro-sowing steps (individual seed placements). The parameter L is encoded in binary as part of the game instance. If a move reaches the L -micro-sow limit before sowing has naturally terminated under R1, the move ends immediately and any unsown seeds remaining in the player’s hand are discarded (removed from play permanently).

Formal modeling note. This truncation rule is a formal device that guarantees move termination in a bounded number of steps. Without it, relay sowing can in principle cycle indefinitely on certain board configurations, which would prevent the game graph from being finite. The truncation semantics are essential to the finite-state formulation and to the EXPTIME upper bound established in this paper.

3.4 Legal Moves

A legal move in state $s = (B, r_A, r_B, T)$ is determined by: (1) selecting a starting pit p on player T ’s side with $b_p \geq 2$; (2) executing the relay-sowing process of R1, applying R2 and R3 capture rules as they arise, subject to the micro-sow limit L of R6; (3) applying R4 mandatory-capture filtering over all candidate starting pits to determine the legal subset. The result of a legal move is a successor state s' in which the player to move has changed to the opponent.

4 Finite-State Game Semantics

4.1 Terminal States

A state s is terminal if the player to move T has no legal starting pit. Under R5, this occurs when every pit on T ’s side contains zero or one seed. The terminal convention is: the player with no legal move loses.

4.2 Infinite Plays and the Finite Game Graph

Under rules R1 through R6 with the lap-cap truncation, every individual move terminates in at most L micro-sows. The state space is finite (proved in Section 5). The game can therefore be represented as a finite directed graph $G = (S, E)$, where nodes are states and edges are legal moves.

However, the finiteness of the state space does not by itself preclude infinite play. Because the game graph may contain directed cycles, a sequence of moves can revisit previously seen states indefinitely. Consequently, the game has three possible outcomes for any initial state:

- **WIN:** player A has a strategy to reach a terminal state where B is to move.
- **LOSS:** player B has a strategy to reach a terminal state where A is to move.
- **DRAW:** neither player can force a terminal state; optimal play leads to infinite cycling.

The existence of the DRAW outcome is not an artifact. A finite graph with cycles admits plays of infinite length. One cannot simply apply ordinary finite-game minimax reasoning, which assumes that every play terminates. A correct analysis must explicitly classify cyclic states.

4.3 WIN/LOSS/DRAW Retrograde Attractor

Definition 2 (Retrograde Attractor). Define the classification function $V : S \rightarrow \{\text{WIN}, \text{LOSS}, \text{DRAW}\}$ by the following iterative process.

Base case. Let W_0 contain all terminal states. For each terminal state s , if the losing player is A, classify s as LOSS (from A’s perspective); if the losing player is B, classify s as WIN.

Iterative step. At stage $k+1$, for each unclassified state s : if $T(s) = A$ and s has at least one successor classified as WIN, classify s as WIN (A can choose a winning move); if $T(s) = A$ and all successors of s are classified as LOSS, classify s as LOSS (every move leads to a loss); symmetrically for $T(s) = B$. Continue until no new classifications are made.

Residual states. All states that remain unclassified after the attractor stabilizes are classified as DRAW.

Lemma 1 (Correctness of the Retrograde Attractor). *The retrograde attractor correctly computes the game-theoretic value of every state in a finite game graph with the terminal-player-loses convention and the draw-on-infinite-play convention.*

Proof. The argument proceeds by establishing that the attractor is a well-defined least fixed point. The state space S is finite, so the iterative process terminates in at most $|S|$ steps. At each stage, a state is classified only if its classification is forced by the classifications of its successors, which is correct by the game-theoretic definition of the value function for alternating reachability games.

For the DRAW classification: a state s remains unclassified if and only if neither player can force termination from s against optimal opponent play. In such a state, both players have strategies to avoid all terminal states indefinitely. Since the graph is finite, any infinite play must cycle, confirming the DRAW interpretation. The attractor's fixed point coincides with the standard definition of winning regions in parity games (with a trivial coloring), which is known to be correct for finite graphs [15]. $\square$

5 State-Space Complexity

5.1 Encoding

An instance of GENERALIZED-BAWO-WIN is specified by: the half-board width n (encoded in binary), the initial board configuration B_0 ($4n$ nonnegative integers, each encoded in binary), the initial reserves $r_{A,0}$ and $r_{B,0}$ (encoded in binary), the lap cap L (encoded in binary), and the starting player T_0 . Let I denote the entire input and $|I|$ its length in bits.

The initial seed total is $S_0 = \sum_i b_{i,0} + r_{A,0} + r_{B,0}$. Since each component is encoded in binary and there are $4n + 2$ components, we have $S_0 \leq (4n + 2) \cdot 2^{|I|}$, so $S_0 \leq 2^{O(|I|)}$. Similarly, $n \leq 2^{|I|}$.

5.2 State Bound

Lemma 2 (State-Space Upper Bound). *The number of well-formed states $|S|$ is at most $2^{\text{poly}(|I|)}$.*

Proof. A well-formed state consists of pit occupancies $(b_0, \dots, b_{4n-1})$, reserves (r_A, r_B) , and the player to move T . The constraint is that the total seed count $S = \sum_i b_i + r_A + r_B$ satisfies $0 \leq S \leq S_0$.

For each fixed total $S \leq S_0$, the number of ways to distribute S seeds among $4n + 2$ containers (pits and reserves) is the number of weak compositions of S into $4n + 2$ parts, given by the binomial coefficient $\binom{S+4n+1}{4n+1}$. Therefore:

$$|S| \leq 2 \cdot \sum_{S=0}^{S_0} \binom{S+4n+1}{4n+1}$$

where the factor of 2 accounts for the player to move. The sum is bounded above by $(S_0 + 1) \cdot \binom{S_0+4n+1}{4n+1}$. Using the standard bound $\binom{a+b}{b} \leq (e(a+b)/b)^b$ and noting that both S_0 and n are at most $2^{O(|I|)}$, we obtain:

$$\log |S| \leq O(n \cdot \log S_0) + O(\log n) + 1 \leq O(2^{|I|} \cdot |I|) = 2^{O(|I|)}.$$

Hence $|S| \leq 2^{\text{poly}(|I|)}$. More precisely, $|S| \leq 2^{2^{O(|I|)}}$, but since the exponent is polynomial in the binary input length (being dominated by products of quantities each at most exponential in $|I|$), the overall bound is exponential in a polynomial of $|I|$. $\square$

Remark. The bound is not tight. The reachable state set S_{reach} may be much smaller than S . However, for the purpose of establishing EXPTIME membership, an upper bound on the size of the state space that can be enumerated suffices.

6 Successor Generation

6.1 Relay Simulation

Given a state s and a candidate starting pit p , the relay-sowing process of R1 is simulated step by step. Each micro-sow (the placement of one seed into one pit) constitutes one step. The relay continues until the final seed of a lap lands in an empty pit or until the lap-cap L is reached.

Since L is encoded in binary, the value of L can be as large as $2^{O(|I|)}$. Therefore, simulating a single move from a single starting pit may require up to $2^{O(|I|)}$ micro-sow steps. Each micro-sow step involves

a constant number of arithmetic operations on integers of size at most $S_0 \leq 2^{O(|I|)}$, each representable in $O(|I|)$ bits. Thus, each micro-sow step takes $O(\text{poly}(|I|))$ time.

6.2 Capture Detection

During relay sowing, capture detection under R2 is checked at each landing in the front row. This requires examining the aligned opponent pit, which is a constant-time operation per landing. The total number of capture checks during a single move is bounded by $L \leq 2^{O(|I|)}$.

6.3 Mandatory-Capture Lookahead

Rule R4 requires that, before a move is accepted, the engine must determine whether any candidate starting pit produces a capture. This requires simulating the move from each candidate starting pit far enough to determine whether a capture occurs. In the worst case, one must simulate the full move (up to L micro-sows) from each of the at most $2n$ candidate pits.

The total cost of mandatory-capture lookahead for a single move decision is therefore $O(n \cdot L \cdot \text{poly}(|I|)) = 2^{O(|I|)}$.

6.4 Binary- L Complexity of Successor Generation

Lemma 3 (Successor Generation). *Given a state s , all legal successors of s can be generated in $2^{\text{poly}(|I|)}$ time.*

Proof. There are at most $2n \leq 2^{|I|+1}$ candidate starting pits. For each candidate pit, simulating the full move takes $O(L \cdot \text{poly}(|I|))$ time. The mandatory-capture lookahead of R4 requires simulating each candidate pit to determine capture status, at the same per-pit cost. Hence the total cost of generating all successors from a single state is $O(n \cdot L \cdot \text{poly}(|I|)) = 2^{O(|I|)} = 2^{\text{poly}(|I|)}$. $\square$

Remark. It is important to note that successor generation is not polynomial in the binary input length. The exponential cost arises from the binary encoding of L . This is consistent with the overall EXPTIME classification.

7 Main Complexity Result

7.1 History-Free Representation

Lemma 4 (History-Free Semantics). *The evolution of a legal move in GENERALIZED-BAWO-WIN is fully determined by the current state $s = (B, r_A, r_B, T)$, the chosen starting pit p , and a fresh per-move micro-sow counter initialized to zero. No additional historical information is required.*

Proof. We verify that the rules R1 through R6 depend only on the current board configuration and move-local state. R1 (relay sowing): the sowing direction is fixed by the board geometry and the current pit index; the decision to relay depends only on whether the landing pit was occupied, which is determined by the current board vector B . R2 (capture): depends only on the landing pit's front-row membership and the aligned opponent pit's occupancy, both determined by B . R3 (capture-relay): depends on whether the landing pit is the designated kichwa pit, determined by the board geometry and the player to move. R4 (mandatory capture): requires scanning candidate pits for capture potential, each of which depends only on B . R5 (no singleton starts): depends only on pit occupancy. R6 (lap cap): depends only on the micro-sow counter, which is initialized to zero at the start of each move.

No rule references any prior move, previous carry history, earlier capture events, or sowing-stage information from a preceding move. The only temporary quantity needed during move execution is the micro-sow counter, which is bounded by L and is reset at each move. Therefore the game's state-transition function is history-free. $\square$

Remark. Lemma 4 is critical for the finite-state formulation. If the rules depended on move history, the state would need to encode that history, potentially inflating the state space beyond the $2^{\text{poly}(|I|)}$ bound.

7.2 Main Theorem

Theorem 2 (EXPTIME Upper Bound). *GENERALIZED-BAWO-WIN under binary- L lap-cap truncation semantics belongs to EXPTIME.*

Proof. We describe a deterministic algorithm that decides GENERALIZED-BAWO-WIN in $2^{\text{poly}(|I|)}$ time.

Step 1: Enumerate the state space. By Lemma 2, the number of well-formed states is $|S| \leq 2^{\text{poly}(|I|)}$. All such states can be enumerated by iterating over all distributions of at most S_0 seeds into $4n + 2$ containers, for each of two players to move. The enumeration takes time proportional to the number of states, hence $O(2^{\text{poly}(|I|)})$.

Step 2: Generate the game graph. For each state $s \in S$, compute all legal successors using the procedure described in Section 6. By Lemma 3, each successor computation takes $2^{\text{poly}(|I|)}$ time. The total time for constructing the game graph is $|S| \cdot 2^{\text{poly}(|I|)} = 2^{\text{poly}(|I|)}$.

Step 3: Compute the retrograde attractor. Apply the WIN/LOSS/DRAW retrograde classification of Definition 2 and Lemma 1. The attractor computation processes each state a constant number of times and performs work proportional to the out-degree of each state. Since the out-degree is at most $2n \leq 2^{|I|}$ and the number of states is $2^{\text{poly}(|I|)}$, the attractor computation takes $2^{\text{poly}(|I|)}$ time.

Step 4: Read the answer. Look up the classification of the initial state. If it is WIN, accept. Otherwise, reject.

Each step runs in $2^{\text{poly}(|I|)}$ time. Hence the total running time is $2^{\text{poly}(|I|)}$, which is EXPTIME. $\square$

7.3 Complexity Qualifications

We state explicitly what has and has not been established.

Proven: GENERALIZED-BAWO-WIN $\in$ EXPTIME.

Not proven:

- PSPACE-hardness of GENERALIZED-BAWO-WIN
- EXPTIME-hardness of GENERALIZED-BAWO-WIN
- PSPACE-completeness of GENERALIZED-BAWO-WIN
- EXPTIME-completeness of GENERALIZED-BAWO-WIN
- PSPACE membership of GENERALIZED-BAWO-WIN
- Complexity classification of historical Bao
- Complexity classification of historical Bawo

EXPTIME membership is currently an upper bound. The exact complexity of GENERALIZED-BAWO-WIN may be lower. In particular, if the game can be solved using only polynomial space (reusing space across the minimax search), the problem may lie in PSPACE. Establishing this would require a different proof technique, such as an alternating polynomial-time simulation.

7.4 Unary- L Variant

If the lap cap L is encoded in unary rather than binary, then $L \leq |I|$. The state-space bound of Lemma 2 remains exponential (it is dominated by the distribution of binary-encoded seed counts across pits). However, the move-simulation cost drops: each move takes at most $O(L \cdot \text{poly}(|I|)) = \text{poly}(|I|)$ time. The mandatory-capture lookahead similarly becomes polynomial.

The overall complexity classification remains EXPTIME under unary- L encoding because the state space is still exponential. The difference is that the per-state work during graph construction becomes polynomial rather than exponential in the input length.

A useful special case arises when $\max_i b_i \leq L$ for every reachable position. In this regime, the lap-cap truncation of R6 never fires, and the relay-sowing process always terminates naturally. This condition is not guaranteed globally; it depends on the specific initial configuration and the dynamics of the rules. When it does hold, the game behaves as if there were no truncation, and seeds are conserved (the total seed count is constant rather than nonincreasing).

8 Computational Verification

The following small-scale computational experiments verify the consistency of the formal model and the retrograde attractor implementation. They are presented as implementation validation, not as evidence for the EXPTIME theorem, which is established by the mathematical proof in Section 7.

8.1 Experiment 1: Sparse Board, $n = 2$

Parameters: $n = 2$, $L = 16$

Initial board: $B = [0, 0, 2, 2, 2, 2, 0, 0]$

Player to move: A

Reachable states: 44

WIN: 21

LOSS: 23

DRAW: 0

The absence of draws is consistent with the small, acyclic structure of this particular game graph.

8.2 Experiment 2: Full Board, $n = 2$

Parameters: $n = 2$, $L = 32$

Initial board: full 2-seed board

Reachable states: 78

WIN: 35

LOSS: 43

DRAW: 0

8.3 Experiment 3: Truncation Event, $n = 4$

Parameters: $n = 4$, $L = 6$

Initial board/reserve total: 11 seeds

After a truncation event triggered by R6:

Seeds retained on board: 7

Seeds discarded: 4

This confirms that the truncation mechanism of R6 operates as specified: when the micro-sow count reaches L , the move terminates and unsown seeds are removed from play.

8.4 Experiment 4: Singleton Terminal Board

Board: $[1, 1, 1, 1, 1, 1, 1, 1]$

Legal starting moves: none

Every pit contains exactly one seed. Under R5, no pit qualifies as a legal start. The position is terminal, and the player to move loses.

9 Relation to Bao and Other Four-Row Mancala Games

The EXPTIME result of Theorem 1 applies to GENERALIZED-BAWO-WIN under rules R1 through R6. It is essential to explain why this result must not be transferred directly to Bao as played in East Africa, or to other four-row mancala games with different rules.

The following documented distinctions between the generalized Bawo model and historical Bao rules affect the formal properties of the game in ways that are relevant to complexity analysis.

Seed recirculation. In many Bao variants, captured seeds are returned to the player's side for continued sowing rather than being permanently removed from play. This affects state conservation: the total seed count may remain constant or even increase on one player's side, rather than being nonincreasing as in the generalized model.

Non-capture relay continuation. In Bao, non-capture moves can continue through relay sowing indefinitely on certain configurations. De Voogt [3, 9] documents the existence of never-ending moves in Bao,

where a sowing sequence cycles without termination. This phenomenon has no analogue in the truncated generalized model (where R6 guarantees termination), and it fundamentally alters the question of whether the game graph is finite.

Direction choice. Some Bao variants allow the player to choose the sowing direction at the start of a move. This introduces additional branching in the game tree and potentially affects the state representation if direction choice interacts with capture rules.

Additional rule stages. Historical Bao often distinguishes between an opening phase (*namua*) and a main phase (*mtaji*), with different rules governing each. The generalized model does not include phase distinctions.

Rule variants and house rules. Both Bao and Bawo exhibit substantial regional variation. Rules regarding capture conditions, kichwa treatment, legal starting pits, and endgame conditions vary across communities and published rule sets.

These differences affect state conservation, move termination, state representation, and successor generation. As a consequence, the finite-state structure that underlies the EXPTIME proof does not necessarily carry over. A game in which moves can be genuinely infinite (without a lap-cap truncation) requires a different formalization, potentially involving infinite-duration game theory or additional termination conventions.

The present EXPTIME result is a theorem about the formal generalized Bawo model, not about Bao as played historically.

10 Limitations

We enumerate the limitations of this work with deliberate thoroughness.

Rule R3 (*kichwa* capture-relay) is a modeling abstraction. It does not reproduce the full ethnographic complexity of kichwa as practiced in any particular regional variant of Bawo. Different formulations of R3 define different formal games.

Rule R6 (lap-cap truncation) is a formal device introduced to guarantee move termination. Historical Bawo and Bao do not include an explicit lap cap. The truncation semantics materially affect the game’s behavior and are essential to the finite-state formulation.

The model does not reproduce every historical Bawo rule or every regional variant. In particular, it does not model phase distinctions (*namua/mtaji*), free direction choice, seed recirculation after capture, or the various house rules documented in the literature.

EXPTIME membership is an upper bound, not a tight classification. No hardness result is established. The exact complexity may be lower; in particular, PSPACE membership has not been ruled out.

The reachable state set S_{reach} is not characterized exactly in general. The state-space bound operates on the larger set S of well-formed states, which may substantially overcount.

The theorem does not classify historical Bao or historical Bawo. Any attempt to transfer the result to these games would require establishing that the relevant rules can be faithfully modeled within the generalized framework, which is a nontrivial and unresolved question.

Succinct encodings of n or other parameters beyond the binary encoding assumed here may require separate analysis.

The computational experiments in Section 8 are implementation validation. They do not constitute evidence for the EXPTIME theorem, which rests entirely on the mathematical proof.

11 Open Problems

We collect the principal open questions arising from this work.

1. Is GENERALIZED-BAWO-WIN PSPACE-hard?
2. Is GENERALIZED-BAWO-WIN EXPTIME-hard?
3. Is GENERALIZED-BAWO-WIN PSPACE-complete?
4. Is GENERALIZED-BAWO-WIN EXPTIME-complete?
5. Can the EXPTIME upper bound be improved, for example to PSPACE?
6. Can the reachable-state complexity be characterized exactly for specific initial configurations?

7. What is the complexity under alternative truncation semantics (for example, cycle detection rather than a lap cap)?
8. What is the complexity without any truncation rule, assuming a draw-on-cycle convention?
9. What is the complexity under Bao-like recirculation of captured seeds?
10. Can the formal model be connected rigorously to ethnographic Bawo rules through a faithful encoding theorem?
11. Can the model support a full history-aware treatment of perpetual moves, as documented in Bao?
12. Can a hardness gadget be constructed to establish a lower bound?
13. What is the effect of adding free direction choice to the generalized model?

12 Conclusion

We have introduced GENERALIZED-BAWO-WIN, a formal finite-state game model inspired by Malawian Bawo, and proved that it belongs to EXPTIME under binary-encoded lap-cap truncation semantics. The proof establishes that the state space is bounded by $2^{\text{poly}(|I|)}$, that successors can be generated in $2^{\text{poly}(|I|)}$ time, and that the WIN/LOSS/DRAW retrograde attractor can be computed within the same time bound. The result is an upper bound; no matching lower bound is established.

The model deliberately abstracts from the full ethnographic complexity of Malawian Bawo and does not attempt to classify Bao as played in East Africa. Establishing the computational complexity of historical four-row mancala games remains a significant open problem that will require both more faithful formal models and new lower-bound techniques.

We hope that this work contributes to the broader program of applying computational complexity theory to traditional combinatorial games from Africa, a domain that has received far less attention than its combinatorial richness deserves.

References

References

- [1] A. de Voogt. A Survey of Mancala in Africa. *Board Game Studies*, 1999.
- [2] A. de Voogt. Distribution of mancala board games: a methodological inquiry. *Board Game Studies*, 2(1):104–114, 1999.
- [3] A. de Voogt. A Limiting Factor in the Distribution of Mancala. In *New Approaches to Board Games Research*, 1995.
- [4] A. de Voogt. Reproduction of Bao positions using a learning algorithm. *Board Game Studies*, 5:63–72, 2002.
- [5] I. Kaimfa. The state-space complexity of Malawian Bawo. Working paper, 2026.
- [6] J. Donkers and J. Uiterwijk. Programming Bao. In *Proceedings of the Computer and Games Conference*, 2000.
- [7] J. Donkers, J. van den Herik, and J. Uiterwijk. Selecting evaluation functions in Bao. In *Advances in Computer Games*, 2003.
- [8] J. Donkers. Searching with Bao. PhD thesis, Universiteit Maastricht, 2002.
- [9] A. de Voogt. The never-ending move in Bao. In *Board Game Studies Colloquium*, 1998.
- [10] D. Lichtenstein and M. Sipser. GO is polynomial-space hard. *Journal of the ACM*, 27(2):393–401, 1980.
- [11] E. Demaine and R. Hearn. Playing games with algorithms: algorithmic combinatorial game theory. In *Games of No Chance 3*, 2009.

- [12] A. Fraenkel and D. Lichtenstein. Computing a perfect strategy for $n \times n$ chess requires time exponential in n . *Journal of Combinatorial Theory, Series A*, 31(2):199–214, 1981.
- [13] J. Robson. The complexity of checkers on an $N \times N$ board. In *Proceedings of FOCS*, 1984.
- [14] S. Reisch. Hex ist PSPACE-vollständig. *Acta Informatica*, 15:167–191, 1981.
- [15] E. Emerson and C. Jutla. Tree automata, mu-calculus, and determinacy. In *Proceedings of FOCS*, 1991.

A Reference Engine Semantics

This appendix records the intended semantics of the game engine used for the computational verification in Section 8. The purpose is to enable independent reproduction of the experimental results.

Move execution. A move from state s with starting pit p proceeds as follows. Initialize a micro-sow counter $c = 0$ and a hand $h = b_p$. Set $b_p = 0$. Enter the sowing loop:

1. Advance to the next pit in the cyclic order. Decrement h by 1 and increment b_{next} by 1. Increment c by 1.
2. If $c = L$, terminate the move immediately. Discard any seeds remaining in hand (h is set to 0; those seeds are removed from play).
3. If $h = 0$ (last seed of the current lap): (a) if the landing pit is on the player’s front row and the aligned opponent front-row pit is nonempty, perform a capture (move opponent pit contents to the player’s reserve); (b) apply the R3 kichwa rule if the landing pit is the designated kichwa pit; (c) if the landing pit was occupied before the seed was dropped (i.e., $b_{\text{landing}} \geq 2$ after the drop), pick up all seeds from the landing pit (set $h = b_{\text{landing}}$, $b_{\text{landing}} = 0$) and continue at step 1; (d) otherwise, the move ends naturally.

Mandatory-capture enforcement. Before accepting a move from starting pit p , the engine simulates moves from all candidate pits with $b_q \geq 2$ to determine which ones produce captures. If any candidate produces a capture and p does not, the move from p is rejected.

B Detailed Counting and Attractor Arguments

B.1 Weak-Composition Bound

The number of weak compositions of a nonnegative integer S into k nonnegative parts is $\binom{S+k-1}{k-1}$. For the generalized Bawo model, $k = 4n + 2$ (the number of pits plus two reserves). For any fixed total $S \leq S_0$, the number of distributions is:

$$\binom{S + 4n + 1}{4n + 1}.$$

Summing over all possible totals $0 \leq S \leq S_0$:

$$\sum_{S=0}^{S_0} \binom{S + 4n + 1}{4n + 1} = \binom{S_0 + 4n + 2}{4n + 2}$$

by the hockey-stick identity. This combinatorial identity gives a slightly tighter form of the bound used in Lemma 2. The logarithm of this quantity is:

$$\log \binom{S_0 + 4n + 2}{4n + 2} \leq (4n + 2) \cdot \log(e \cdot (S_0 + 4n + 2)/(4n + 2)).$$

Since $n \leq 2^{|I|}$ and $S_0 \leq 2^{O(|I|)}$, this logarithm is at most $2^{O(|I|)}$, confirming the exponential state-space bound.

B.2 Attractor Termination

The retrograde attractor of Definition 2 terminates in at most $|S|$ iterations. At each iteration, at least one previously unclassified state is classified (as WIN or LOSS). When no new classification occurs, the attractor has stabilized. All remaining unclassified states are classified as DRAW.

The total work performed by the attractor is $O(|S| + |E|)$, where $|E|$ is the number of edges in the game graph. This is because each state is inspected at most a constant number of times (when one of its predecessors changes classification), and each edge is traversed at most once in the reverse direction.

Since $|S| \leq 2^{\text{poly}(|I|)}$ and each state has out-degree at most $2n$, we have $|E| \leq 2n \cdot |S| \leq 2^{\text{poly}(|I|)}$. Hence the attractor computation runs in $2^{\text{poly}(|I|)}$ time, consistent with the EXPTIME bound of Theorem 1.