The Shannon Number and the Nature of "Best Moves" in Modern Chess Engines

Marco Cekada

October 2025

Abstract

In 1950 Claude Shannon, for the first time, tried to caratterize chess as a game-tree complexity, on the order of 10^{120}. This is known as the *Shannon Number* (Shannon, 1950) and will open up new research on artificial intelligence. This paper will revisit the maths behind the game. It's known as the 10th ply generates $\approx 6.935 \times 10^{13}$ possible games and starting from this we are going to explains how contemporary engines compute *best moves* while respecting time and calculation constraints. We compare symbolic engines (minimax with α–β pruning, transposition tables, quiescence search) and subsymbolic systems (deep neural evaluation and Monte Carlo Tree Search), and outline hybrid directions. The goal is also to give a precise meaning to the status of "best" move taking into account limited computing power and prior knowledge. With a concrete case study we will compare AlphaZero's policy-guided search to Stockfish's depth-first enumeration. Throughout, we are going to connect search to theoretic notions and computational complexity, clarifying how can a fast and high-quality decision-making being possible despite exponential growth (Campbell, Hoane, & Hsu, 2002; Silver et al., 2018).

1 Introduction

Chess game is an important sample for the study of decision making in the presence of a combinatorial explosion. Even modest horizons of analysis generate research spaces beyond the scope of exhaustive enumeration, yet modern engines can make near-optimal moves in a few milliseconds. The main factors are:
(i) selective expansion (search) with strong pruning;
(ii) high quality assessment (crafted or learned);
(iii) reuse of the previous calculation with hashing;
(iv) allocation of computation under time control.
These mechanisms lead to the program sketched by Shannon (1950) and refined through decades of AI research (Russell & Norvig, 2021).

2 The Shannon Number and Combinatorial Growth

Let b be the average branching factor and m be the number of layers so the approximate Shannon model is

$$N \approx b^m. \tag{1}$$

Using $b \approx 35$ and $m \approx 80$ gives $N \approx 35^{80} \approx 10^{120}$ (Shannon, 1950). While the per-ply b is nonstationary (openings often 20–30, middlegames 30–40, endgames 5–15), any realistic game produces exponential growth.

3 Mathematical Structure of Early Growth

If b_i be is the average count of legal moves in ply i, the number of distinct *game sequences* after n plies is:

$$G(n) = \prod_{i=1}^{n} b_i. \tag{2}$$

Under a constant model $b_i \equiv b$,

$$G(10) = b^{10}. \tag{3}$$

With $b = 35$,

$$G(10) = 35^{10} = 6.935\,285\,971\,241\,7 \times 10^{13}. \tag{4}$$

A phase-aware profile (opening, middlegame, transition) provides

$$G(10) = 20^2 \cdot 35^6 \cdot 25^2 = 5.502\,734\,375 \times 10^{13}.$$

The log recording clarifies the slopes:

$$\log G(n) = \sum_{i=1}^{n} \log b_i \overset{b_i \equiv b}{=} n \log b. \tag{5}$$

Therefore, each additional ply multiplies the possibilities by approximately b.

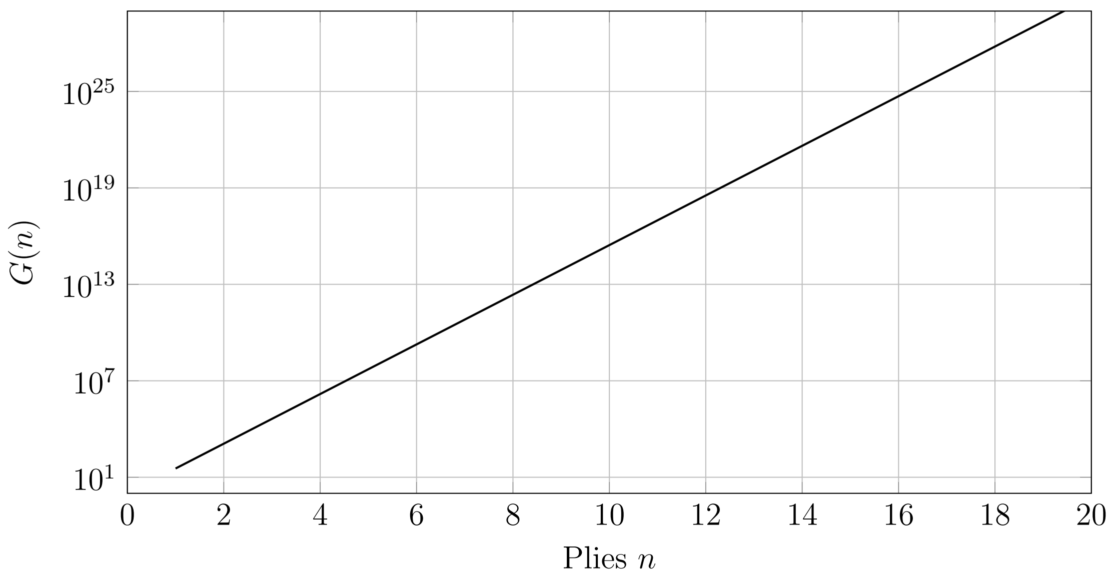

Figure 1: Exponential growth of $G(n) = 35^n$ (log scale).

3.1 Positions vs. Sequences

Sequences overestimate transpositions. Let $\mathcal{P}_n$ be the set of *positions* reachable by depth n. Then $|\mathcal{P}_n| \ll G(n)$, yet estimates still reach $\sim 10^{46}$ for overall legal positions. [1]

3.2 Complexity and Lower Bounds

With a perfect α–β ordering, the node count after depth d is approximately $\Theta(b^{d/2})$, still exponential. Stirling-type approximations for branching profiles show that reductions in b (with pruning or policy guidance) multiply with depth.

4 Computational and Information-Theoretic View

Search converts uncertainty (entropy) into decisional certainty. Let H be the entropy on the game's outcomes, conditioned by current knowledge (ply-bound search and evaluation parameters). Each node expands the evidence; pruning corresponds to early stopping, when the posterior mass concentrates and neural priorities further reduce the initial entropy, biasing expansions toward high-reward actions. This view is consistent with bounded rationality: maximizing the quality of decisions given computational/time constraints (Russell & Norvig, 2021).

[1]The upper/lower bounds depend on legality, repetition, and symmetry constraints.

5 Classical Engines

5.1 Minimax and α–β

Define the value of position p in a perfect game as $V(p)$. The one-step backup is

$$V(p) = \max_{m \in M(p)} \min_{r \in R(m)} V(r). \tag{6}$$

α–β pruning maintains (α, β) bounds on provable values and discards branches that cannot improve them, often producing orders of magnitude savings (Campbell et al., 2002).

5.2 Evaluation and Quiescence

At leaves, engines apply

$$\text{Eval}(p) = \sum_{j=1}^{k} a_j f_j(p), \tag{7}$$

with characteristics for material, mobility, king safety, pawn structure, space, and initiative; the coefficients a_j are adjusted via self-play and testing. Quiescence extends the noisy tactical lines to avoid the "horizon effect".

5.3 Transpositions and Hashing

Zobrist hashing assigns a nearly unique key to each position; transposition tables store $(\text{Eval}, \text{depth}, \alpha, \beta)$ tuples to prevent recalculation (Zobrist, 1970). Iterative deepening improves the move ordering (which in turn improves α–β).

6 Neural and Hybrid Engines

6.1 Deep RL with MCTS

A deep network $f_\theta(s)$ returns a policy $P_\theta(\cdot|s)$ and a value $V_\theta(s)$. Self-reproduction minimizes

$$\mathcal{L}(\theta) = (z - V_\theta(s))^2 - \pi^\top \log P_\theta(\cdot|s) + \lambda \|\theta\|^2, \tag{8}$$

where $z \in \{-1, 0, 1\}$ is the outcome of the game and π is the MCTS-improved policy (Silver et al., 2018). Node selection uses

$$UCB = Q + c_{\text{puct}}\, P \, \frac{\sqrt{N_{\text{parent}}}}{1 + N_{\text{child}}}, \tag{9}$$

balancing the exploitation (Q) and the exploration (P).

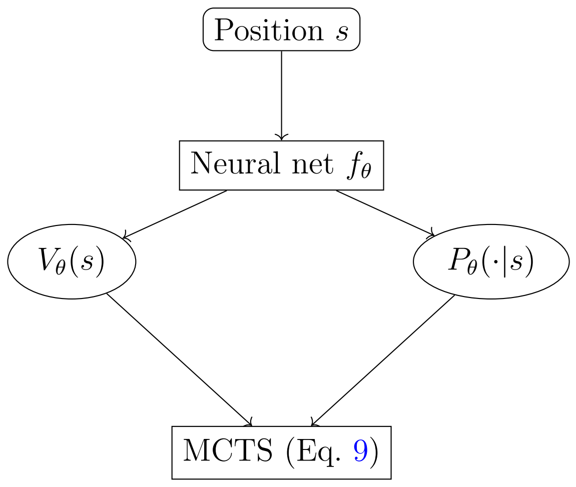
$$\begin{tikzcd}
 & \text{Position } s \arrow[d] & \\
 & \text{Neural net } f_\theta \arrow[dl] \arrow[dr] & \\
V_\theta(s) \arrow[dr] & & P_\theta(\cdot|s) \arrow[dl] \\
 & \text{MCTS (Eq. 9)} &
\end{tikzcd}$$

Figure 2: Neural assessment guiding MCTS through policy priorities and value estimates.

6.2 Comparison and Hybrids

Classical engines are highly tactically efficient with precise search and handcrafted heuristics; neural engines, on the other hand, generalize with learned priors and implicit pattern recognition. Hybrid systems inject V_θ (or P_θ) into the α–β search, or use NN-guided move sorting, combining symbolic precision with learned guidance.

7 Case Study: AlphaZero vs. Stockfish - A Concrete Illustration

The above theoretical arguments can be based on a concrete scenario taken from the first series of games between *AlphaZero* and *Stockfish* (Silver et al., 2018; Sadler & Regan, 2019). AlphaZero, trained only by autoplay for 24 hours, defeated Stockfish 28-0 with 72 draws. This was a remarkable result because it demonstrated how learning-based reasoning can outperform exhaustive computation.

7.1 Opening Context: The Nimzo-Indian Defence

Consider the classical sequence:

$$1.\, d4\, Nf6 \;\; 2.\, c4\, e6 \;\; 3.\, Nc3\, Bb4 \;\; 4.\, e3\, O-O \;\; 5.\, Bd3\, d5,$$

a well known position in the *Nimzo-Indian Defence*. After five moves each (ten plies), the Shannon model predicts approximately 6.910^{13} possible distinct continuations of the game and even at this depth, an exhaustive calculation is already impossible.

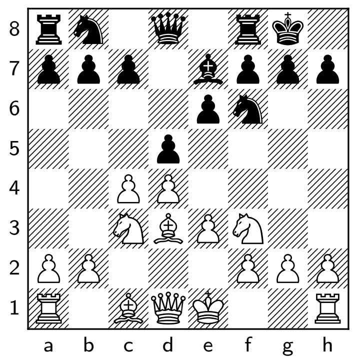

Figure 3: The Nimzo-Indian position after 1.d4 Nf6 2.c4 e6 3.Nc3 Bb4 4.e3 O-O 5.Bd3 d5.

7.2 Stockfish's Approach: Symbolic and Exhaustive

At this point, Stockfish, which computes more than 6.5×10^6 nodes per second, evaluates millions of possible continuations in a few seconds. Its evaluation function, following (7), will linearly combine the explicit positional factors:

$$Eval(p) = a_1 M + a_2 P + a_3 K + a_4 S,$$

where M indicates the material balance, P the pawn structure, K the king's safety, and S the mobility. The coefficients a_i are adjusted empirically through autonomous game optimization.

In this position, Stockfish will likely recommend

$$\ldots c5$$

- a central disruption consistent with its material-centric heuristics. The engine's search horizon (typically 40-50 layers) explores short-term tactics in depth, but cannot find long-term strategic compensation. Its reasoning is therefore *symbolic*: explicit rules govern each evaluation stage, and the efficiency of pruning relies on syntactic ordering.

7.3 AlphaZero's Approach: Neural and Predictive

AlphaZero, on the other hand, does not perform an artisanal evaluation. Its neural function $f_\theta(s)$ maps the board representation s into a scalar value $V_\theta(s)$ and a policy distribution $P_\theta(\cdot|s)$:

$$f_\theta(s) = \sigma(W_2\,\mathrm{ReLU}(W_1 s + b_1) + b_2).$$

During play, P_θ biases Monte Carlo Tree Search expansions via (9), focusing computation on promising continuations.

In this position, the network prefers

$$\ldots b6,$$

preparing a queenside expansion and pressure on the dark square. Although short-term heuristics rate the move slightly lower, AlphaZero's previous learnings predict a favorable long-term structure. The head value V_θ assigns a small but stable advantage, reflecting positional understanding rather than tactical calculation.

7.4 Comparative Dynamics

Stockfish explores broadly and symmetrically, while AlphaZero selectively assigns depth to branches aligned with its policy priorities. We can try to formalize the effective branching factor b_{eff} as

$$b_{\text{eff}} = \sum_a P_\theta(a|s),$$

which, under a sharp peak in P_θ, yields $b_{\text{eff}} \ll b$. Thus, neural guidance reduces the practical size of the explored tree by orders of magnitude with a probabilistic analog of α-β pruning.

7.5 Interpretation and Connection to Shannon

The contrast explains Shannon's principle of bounded rationality within an exponential domain. After ten passes, there are $\sim 7 \times 10^{13}$ possible sequences, yet AlphaZero only needs to consider a few thousand more promising ones thanks to its learned priorities and its value guidance. Both engines achieve the same theoretical goal of approximating minimax optimality, but through different epistemic paths:

- **Stockfish:** exhaustive symbolic reasoning, optimizing within a set of discrete rules.
- **AlphaZero:** statistical inference, compressing experiential data into a latent strategy.

In Shannon terms, the engine converts raw combinatorial entropy into actionable insights. Search and evaluation act as entropy-reducing transformations: Stockfish does this through rule-based pruning, while AlphaZero does it through learned compression of the a priori search.

This case study concretely demonstrates how modern AI realizes Shannon's 1950 vision: machines that move beyond brute-force enumeration to operate through inference, abstraction, and learning.

8 Conclusion

The Shannon framework (1) and depth-first analysis (2)–(3) explain why naive enumeration is not feasible: even $n = 10$ produces $\sim 10^{14}$ sequences. The engines succeed thanks to *structure*: pruning, evaluation, and reuse. Classical systems effectively reduce branching via α–β and superior move ordering; neural systems apply preconditions to the search distribution with P_θ and compress strategic knowledge into V_θ. Both implement bounded optimality under time constraints (Russell & Norvig, 2021).

From a theoretical perspective, search and evaluation progressively reduce the a posteriori entropy of results; pruning corresponds to an early conclusion once the credible intervals are reduced. Hybrid engines, already competitive in practice, ensure the best of both worlds by combining precise tactical computation where it matters with neural guidance where the structure is fine. As the training corpora (self-play) and the computational scale increase, priorities become more refined and the effective branching factor decreases with depth.

Open directions include: (i) interpretable neural evaluation linking V_θ and human concepts; (ii) learning-based pruning criteria with theoretical guarantees; (iii) integrating proven tablebases and endgame oracles into learned systems; (iv) generalizing these methods beyond chess to theorem proving, scientific discovery, and control, domains where decision quality must emerge based on exponential chance (Campbell et al., 2002; Silver et al., 2018).

References

Campbell, M., Hoane, A. J., & Hsu, F.-h. (2002). Deep blue. *Artificial Intelligence*, *134*(1–2), 57–83. doi: 10.1016/S0004-3702(01)00129-1

Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Hoboken, NJ: Pearson.

Sadler, M., & Regan, N. (2019). *Game changer: Alphazero's groundbreaking chess strategies and the promise of ai.* Alkmaar, NL: New In Chess.

Shannon, C. E. (1950). Programming a computer for playing chess. *Philosophical Magazine*, *41*(314), 256–275.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*(6419), 1140–1144. doi: 10.1126/science.aar6404

Zobrist, A. L. (1970). *A new hashing method with application for game playing* (Tech. Rep. No. Technical Report 88). University of Wisconsin.