

What's New in ShashChess 38



ShashChess 38 represents a significant evolution compared to Stockfish, introducing strategic and technical innovations that distinguish it as a derived engine rather than a clone. The Shashin zones, the WDL model, and a dynamic approach to handling difficult positions make it unique in the landscape of chess engines.

Detailed Technical Analysis of Innovations Introduced

Isolation of the Shashin Framework

Originality

The Shashin framework introduces advanced positional zone management, now isolated into a dedicated package to improve modularity, maintainability, and scalability.

Implementation

Separated files

The framework is organized into files such as ``shashin_helper.h``, ``shashin_manager.cpp``, and ``shashin_types.h``, managing zone classification, state updates, and integration with the engine.

Defined interfaces

Clear APIs like ``getShashinRange``, ``initShashinValues``, and ``updateShashinValues`` allow for easy interaction with the framework.

Independence

Although integrated into the search pipeline, the Shashin framework is autonomous, making it easily upgradable.

Advantages

Modularity

Enables improvements to the Shashin framework without affecting the rest of the engine.

Testability

Allows for separate testing to ensure precise zone determination.

Scalability

Facilitates the integration of new features or zones.

Introduction of the WDL (Win-Draw-Loss) Model

Innovation

The WDL model surpasses the limitations of the simple win probability, adding granularity and precision in positional classification. Based on the values (w, d, l), the model distinguishes positions with different characteristics even at the same win probability.

Implementation

WDL Calculation

- w: Percentage of expected wins;
- d: Percentage of expected draws;
- l: Percentage of expected losses;
- Win Probability = $w + d/2$;

- **Zone Mapping**:

- Zones like High Tal, Capablanca, Middle Petrosian, etc., are defined by symmetric ranges respecting positional balance (e.g., Petrosian ↔ Tal).

- Capablanca acts as the positional center, represented by (w, d, l) = (0, 100, 0) with Win Probability = 50.

- **Symmetry maintained**:

- High Petrosian Win = High Tal Loss.

- Capablanca-Petrosian Loss = Capablanca-Tal Win.

- **Dynamic Integration**:

- `WDLModel::get_wdl(value, rootPos)` determines the probabilities of wins, draws, and losses, classified through `getShashinRange`.

Advantages

Precision

Accurately distinguishes between positions with similar win probability but different characteristics.

Adaptability

Allows the engine to respond effectively to both tactical and strategic needs.

Resilience

Optimally handles chaotic or critical positions.

Initialization and Update of Shashin State

Guiding Principle

The Shashin approach is inspired by the principle of maximum resilience in thermodynamics, suggesting starting with an aggressive approach (e.g., High Tal) and progressively adapting to the position.

Implementation

State Initialization

- During ``initShashinValues``, the engine begins with an aggressive state (e.g., High Tal or Chaos) to explore tactical opportunities.
- The ``RootShashinState`` structure tracks key parameters such as depth, current zone, legal move count, material, and king safety.

Dynamic Updates

During ``updateShashinValues``, the framework recalculates the Shashin zone at every search depth, adapting the strategy, based on the following table:

WDL Range (W, D, L)	Shashin Position's Type	Win Probability Range	Informator Symbols	Description
[0, 3], [0, 4], [96, 100]	High Petrosian	[0, 5]	-+	Winning: a decisive advantage, with the position clearly leading to victory.
[4, 6], [5, 8], [89, 95]	High-Middle Petrosian	[6, 10]	-+ \ -/+	Decisive advantage: dominant position and likely winning.
[7, 9], [9, 12], [80, 87]	Middle Petrosian	[11, 15]	-/+	Clear advantage: a substantial positional advantage, but a win is not yet inevitable.
[10, 12], [13, 16], [73, 79]	Middle-Low Petrosian	[16, 20]	-/+ \ =/+	Significant advantage: strong edge
[13, 15], [17, 39], [66, 71]	Low Petrosian	[21, 24]	=/+	Slight advantage with a positional edge, but no immediate threats.
[0, 30], [40, 99], [31, 64]	Chaos: Capablanca-Petrosian	[25, 49]	↓	Opponent pressure and initiative: defensive position.
[0, 0], [100, 100], [0, 0]	Capablanca	[50, 50]	=	Equal position. Both sides are evenly matched, with no evident advantage.
[30, 64], [40, 99], [0, 30]	Chaos: Capablanca-Tal	[51, 75]	↑	Initiative: playing dictation with active moves and forcing ideas.
[65, 71], [17, 39], [13, 15]	Low Tal	[76, 79]	+/=	Slight advantage: a minor positional edge, but it's not significant.
[72, 78], [13, 16], [10, 12]	Middle-Low Tal	[80, 84]	+/= \ +/-	Slightly better, tending toward a clear advantage. The advantage is growing, but the position is still not decisive.
[79, 87], [9, 12], [7, 9]	Middle Tal	[85, 89]	+/-	Clear advantage: a significant edge, but still with defensive chances.
[88, 95], [5, 8], [4, 6]	High-Middle Tal	[90, 94]	+/- \ +-	Dominant position, almost decisive, not quite winning yet, but trending toward victory.
[96, 100], [0, 4], [0, 3]	High Tal	[95, 100]	+-	Winning: a decisive advantage, with victory nearly assured.
In particular, [33, 33], [33, 33], [33, 33]	Chaos: Capablanca-Petrosian-Tal	(Unclassified)	∞	Total chaos: unclear position, dynamically balanced, with no clear advantage for either side and no clear positional trends.

N.B. The wdl model also takes into account the history of the position at which a move has been calculated. So, it's more effective than the cp.

Chaos Management

In uncertain positions (e.g., Chaos: Capablanca-Petrosian-Tal), the engine explores a wide range of moves, reducing aggressive pruning.

Advantages

Progressiveness

Gradual transition from tactical to strategic play, enhancing dynamic understanding.

Resilience

Optimal handling of unstable or complex positions.

Continuous Adaptation

Strategic evolution during the game.

Added Value Compared to Stockfish

ShashChess is not a clone of Stockfish but a derived engine introducing significant innovations:

Dynamic Flexibility

The Shashin framework allows the engine to dynamically adapt between tactical (e.g., Tal) and strategic (e.g., Petrosian) approaches.

Precision in Calculation

The WDL model improves the engine's ability to distinguish between positions with similar characteristics, optimizing the handling of critical situations.

Chaos Management

Shashin's approach ensures superior management of complex positions compared to Stockfish's standard pruning.

Performance in Long Time Controls^{**}:

Thanks to resilience and adaptability, ShashChess excels in long-time matches while maintaining high competitiveness in fast games.

Benchmark Results

ShashChess was subjected to rigorous testing to validate its improvements:

LTC Match on 10 "Pepite" Positions

ShashChess won 4-3 against Stockfish, with 17 draws, demonstrating its superior handling of complex and tactically sharp positions: <https://github.com/amchess/ShashChess/wiki/Match>

Suite of 128 Hard Positions

ShashChess solved 22 more positions than Stockfish, underscoring its enhanced ability to manage difficult scenarios effectively: <https://github.com/amchess/ShashChess/wiki/Battery>

Conclusion

The integration of the Shashin framework, the WDL model, and dynamic zone management represents an innovative and exceptional idea. ShashChess is a flexible and adaptable engine, capable of handling any positional scenario with strategic intelligence. These innovations distinguish it as a significant contribution to the chess community and a truly original and innovative derivation of Stockfish.