TRADING PHILOSOPHY

✅ Target: 10-15% monthly return with maximum 5% drawdown

✅ Risk/Reward Ratio: Minimum 1:2 (ideal 1:3)

✅ Success Rate: Minimum 60% (target 70%)

✅ Daily Trades: 3-5 maximum (avoid over-trading)

✅ Traded Pairs: XAUUSD primary, then EURUSD, GBPUSD, USDJPY

📈 Follow main trend (H4/D1)

🎯 Enter on retracements (H1/M15 structure)

🔒 Strict risk management with 50/50 strategy

⏰ Trade during London session (8AM-5PM Paris time)

🚫 No trading during major news (High Impact)

STRATEGY OPRIMIZATION

TIMEFRAME CONFIGURATION

// Recommended configuration:

Trend Timeframe: H4 (directional trend)

Signal Timeframe: H1 (entry setup)

Entry Timeframe: M15 (precise timing)

Confirmation: M5 (additional filter)

TECHNICAL PARAMETERS TO OPTIMIZE

struct OptimizationParams {

// Moving Averages

int ema\_fast\_period = {9, 13, 21, 34}; // Test range

int ema\_slow\_period = {21, 34, 55, 89}; // Test range

int ema\_trend\_period = {89, 144, 233}; // Test range

// Momentum

int rsi\_period = {7, 14, 21}; // Test RSI

double rsi\_oversold = {28, 30, 32}; // Optimal levels

double rsi\_overbought = {68, 70, 72}; // Optimal levels

// Volatility

int atr\_period = {7, 14, 21}; // Test ATR

double atr\_multiplier\_sl = {1.5, 2.0, 2.5}; // Stop Loss

double atr\_multiplier\_tp = {2.0, 2.5, 3.0}; // Take Profit

// Filters

double min\_adx = {20, 25, 30}; // Trend strength

int min\_confluence\_points = {3, 4, 5}; // Confluence points

double min\_signal\_confidence = {0.6, 0.7, 0.75}; // Confidence

};

ENTRY CONDITIONS TO OPTIMIZE

// Priority 1: Multi-timeframe confluence (TO OPTIMIZE)

int CalculateConfluence(ENUM\_ORDER\_TYPE direction) {

// Score should be optimized for each pair

1. H4 trend alignment = 25% of score

2. H1 RSI momentum = 20% of score

3. H1 MACD crossover = 20% of score

4. Price vs EMA M15 = 15% of score

5. Volume confirmation = 10% of score

6. Volatility (ATR) = 10% of score

// Minimum score required: TO OPTIMIZE (60-75%)

}

RISK MANAGEMENT PARAMETERS

struct RiskRequirements {

// RISK PER TRADE

double base\_risk\_per\_trade = 1.0%; // Max 1% per trade

double max\_position\_risk = 2.0%; // Max 2% per position

double risk\_adjustment\_factor = 0.8; // Reduction after loss

// DAILY LIMITS

double daily\_loss\_limit = 3.0%; // Daily stop at 3%

double daily\_profit\_target = 5.0%; // Daily target at 5%

int max\_daily\_trades = 5; // Max 5 trades/day

int max\_consecutive\_losses = 3; // Stop after 3 losses

// 50/50 STRATEGY - TO OPTIMIZE

bool enable\_50\_50\_strategy = true;

double tp\_fixed\_ratio = {0.4, 0.5, 0.6}; // Fixed TP ratio

double tp\_fixed\_multiplier = {1.0, 1.25, 1.5}; // TP multiplier

double trailing\_start\_rr = {1.0, 1.25, 1.5}; // Trailing start

double trailing\_atr\_mult = {1.0, 1.25, 1.5}; // Trailing distance

// DYNAMIC RISK BASED ON VOLATILITY

bool adjust\_risk\_by\_volatility = true;

double max\_risk\_reduction = 0.5; // Maximum reduction 50%

double volatility\_threshold = 2.0; // ATR multiplier

};

POSITION SIZING ALGORITHM

// TO IMPLEMENT: Dynamic position sizing

double CalculateDynamicPositionSize() {

// Factor 1: Market volatility (ATR)

double atr\_factor = NormalizeATR(current\_atr);

// Factor 2: Trend strength (ADX)

double trend\_factor = CalculateTrendStrength();

// Factor 3: Signal confluence (0-1)

double confluence\_factor = signal\_confidence;

// Factor 4: Recent performance (win rate)

double performance\_factor = CalculateRecentPerformance();

// Final calculation

double base\_size = standard\_position\_size;

double adjusted\_size = base\_size \* atr\_factor \* trend\_factor

\* confluence\_factor \* performance\_factor;

// Apply min/max limits

return Clamp(adjusted\_size, min\_lot, max\_lot);

}

EXECUTION OPTIMIZATION

EXECUTION REQUIREMENTS

struct ExecutionRequirements {

// TIMING

int max\_order\_delay\_ms = 500; // Max 500ms delay

bool use\_iceberg\_orders = false; // For large volumes

int partial\_fill\_timeout = 2000; // Timeout 2 seconds

// SLIPPAGE & SPREAD

double max\_slippage\_points = 3.0; // Max 3 pips slippage

double max\_allowed\_spread = {

"XAUUSD": 50.0, // 50 pips max

"EURUSD": 20.0, // 20 pips max

"GBPUSD": 25.0, // 25 pips max

"USDJPY": 30.0 // 30 pips max

};

// RETRY LOGIC

int max\_execution\_retries = 3; // Max 3 attempts

int retry\_delay\_ms = {100, 250, 500}; // Progressive delays

bool cancel\_on\_spike = true; // Cancel on spike

// SMART ORDER TYPES

bool use\_limit\_entries = true; // Limit entries

double limit\_offset\_pips = 5.0; // Offset 5 pips

bool use\_trailing\_on\_entry = false; // Trailing on entry

};

MARKET FILTERS TO IMPLEMENT

// NEW: Advanced filters to add

class AdvancedMarketFilters {

// 1. Liquidity filter

bool CheckLiquidity(string symbol, double volume) {

// Check market volume

// Check bid/ask spread

// Check market depth

return liquidity\_adequate;

}

// 2. Correlation filter

bool CheckCorrelation(string symbol, ENUM\_ORDER\_TYPE direction) {

// Avoid trades in same direction on correlated pairs

// EURUSD and GBPUSD positively correlated

// USDJPY and XAUUSD negatively correlated

return correlation\_acceptable;

}

// 3. Extreme momentum filter

bool CheckExtremeMomentum(string symbol) {

// Detect abnormal movements (news, manipulation)

// Use RSI, candle speed, gaps

return momentum\_normal;

}

};

MONITORING AND REPORTING

MONITORING DASHBOARD

// TO DEVELOP: Web-based monitoring interface

struct DashboardRequirements {

// REAL-TIME

bool enable\_web\_socket = true; // Real-time updates

int update\_interval\_ms = 1000; // Update every 1s

// METRICS TO DISPLAY

string[] display\_metrics = {

"Real-time Equity & Balance",

"Current and maximum drawdown",

"Open position (price, SL, TP)",

"Generated signals (confidence, direction)",

"Daily/weekly performance",

"Trading statistics (win rate, R:R)",

"Risk exposure per pair",

"Integrated news calendar"

};

// ALERTS

string[] alert\_triggers = {

"Drawdown > 3%",

"Abnormal spread detected",

"Major news in 15min",

"Broker connection lost",

"Margin < 20%",

"5 consecutive losing trades"

};

// REPORTING

bool generate\_daily\_report = true; // Daily PDF report

bool enable\_telegram\_alerts = true; // Telegram alerts

bool cloud\_backup = true; // Cloud backup

};

SECURITU AND ROBUSTNESS

SECURITU REQUIREMENTS

struct SecurityRequirements {

// BUG PROTECTION

bool enable\_circuit\_breakers = true; // Circuit breakers

double emergency\_stop\_drawdown = 10.0%; // Stop at 10% DD

// BACKUP

bool auto\_save\_state = true; // Auto save state

int save\_interval\_minutes = 5; // Every 5 minutes

bool cloud\_state\_backup = true; // Cloud backup

// CONNECTION

bool auto\_reconnect = true; // Auto reconnect

int max\_reconnection\_attempts = 5; // Max 5 attempts

int reconnection\_delay = 30000; // 30 seconds delay

// VALIDATION

bool pre\_trade\_validation = true; // Pre-trade validation

bool post\_trade\_validation = true; // Post-trade validation

bool sanity\_checks = true; // Sanity checks

};