

V1N1 LONNY — How to optimize the per-symbol parameters

Step-by-step guide for the MetaTrader 5 Strategy Tester.

Understanding the mechanism (read this first)

V1N1 LONNY is a multi-symbol EA. For that reason, the parameters that depend on the instrument are not inputs: they are stored in an internal table inside the `InitSymbolParamsTable()` function:

```
SetSymbolParams(i++, "EURUSD", 60, 6, 0.16, 14, 6);
//           ^           ^ ^   ^   ^   ^
//           pattern  ADR  H1  SAR  MACD  Asian
```

If those values were regular inputs, the same set would be applied to every symbol at once — which makes no sense, since EURUSD and XAUUSD call for different settings.

The optimization therefore follows this logic:

1. You temporarily expose the 5 parameters as inputs (the block is already written and commented out in the code);
2. You bind those inputs to the table row of a single symbol;
3. You optimize that symbol alone in the Tester;
4. You write the result back into the table as fixed numbers;
5. You repeat for the next symbol.

The 5 parameters

Optimization input	Table field	Allowed values	What it controls
<code>OPT_PARAM_ADR_PERIOD</code>	<code>adrPeriod</code>	6, 12, 24, 42, 60	ADR/ATR lookback on D1 (in days)
<code>OPT_PARAM_PERIOD_H1</code>	<code>periodH1</code>	0, 6, 14, 22	Indicator period in hours (0 = dynamic, derived from the symbol's session)
<code>OPT_PARAM_STEP_SAR</code>	<code>stepSAR</code>	2, 9, 16	PSAR step (the input is ×100 — see Step 8)
<code>OPT_PARAM_MACD_SIGNAL_H1</code>	<code>macdSignalH1</code>	9, 14, 21	MACD Signal in hours (Fast and Slow are derived from it)
<code>OPT_PARAM_ASIAN_RANGE_BARS_H1</code>	<code>asianRangeBarsH1</code>	6, 8, 10	Asian range window in hours

Total: $5 \times 4 \times 3 \times 3 \times 3 = 540$ combinations. That is a small space — you can use the complete algorithm, no genetics needed.

Step 1 — Make a working copy of the EA

Duplicate the file, for example as `V1N1_LONNY_OPT.mq5`.

Every change made in Steps 2 and 3 is temporary and exists only for optimization. Keeping the original EA untouched prevents you from ever running a live account on a build that still has the optimization inputs wired in.

Step 2 — Uncomment the "OPTIMIZATION SETTINGS" block

Right above the `INPUT PARAMETERS` section there is a block enclosed in `/*` and `*/`. Remove both comment markers:

```
input group "OPTIMIZATION SETTINGS"
input ENUM_ADR_PERIOD OPT_PARAM_ADR_PERIOD = DEFAULT_ADR_PERIOD;
input ENUM_PERIOD_H1 OPT_PARAM_PERIOD_H1 = DEFAULT_PERIOD_H1;
input ENUM_STEP_SAR OPT_PARAM_STEP_SAR = STEP_SAR_002;
input ENUM_MACD_SIGNAL OPT_PARAM_MACD_SIGNAL_H1 = DEFAULT_MACD_SIGNAL_H1;
input ENUM_ASIAN_RANGE_BARS OPT_PARAM_ASIAN_RANGE_BARS_H1 =
DEFAULT_ASIAN_RANGE_BARS_H1;

double optParamStepSAR = OPT_PARAM_STEP_SAR / 100.0;
```

This creates the OPTIMIZATION SETTINGS group in the Tester's Inputs tab. The `optParamStepSAR` variable already converts the PSAR step to the decimal scale used internally.

Step 3 — Bind the table to the inputs (one symbol at a time)

Inside `InitSymbolParamsTable()`, find the row of the symbol you are going to optimize and replace the 5 numbers with the optimization variables.

Before (EURUSD, as an example):

```
SetSymbolParams(i++, "EURUSD", 60, 6, 0.16, 14, 6);
```

After:

```
SetSymbolParams(i++, "EURUSD", OPT_PARAM_ADR_PERIOD, OPT_PARAM_PERIOD_H1,
optParamStepSAR, OPT_PARAM_MACD_SIGNAL_H1, OPT_PARAM_ASIAN_RANGE_BARS_H1);
```

Change one row at a time only. If you replace several, those symbols will share the same set and the result stops being per-symbol.

If the symbol is not in the table yet

Add a new row and increase the `ArrayResize` at the top of the function, otherwise the entry is silently discarded:

```
ArrayResize(g_symbolParamsTable, 20); // was 19
...
SetSymbolParams(i++, "GBPUSD", OPT_PARAM_ADR_PERIOD, OPT_PARAM_PERIOD_H1,
optParamStepSAR, OPT_PARAM_MACD_SIGNAL_H1, OPT_PARAM_ASIAN_RANGE_BARS_H1);
```

Two notes about how the pattern is matched:

- The comparison is by substring, not by exact equality (`StringFind`). The pattern "EURUSD" matches EURUSD, EURUSD.m, EURUSDx, EURUSD_raw and so on — there is no need to include your broker's suffix.
- The first matching pattern in table order wins. When creating new patterns, avoid one being contained in another.

Step 4 — Compile

Press **F7** in MetaEditor. Confirm 0 errors before moving on.

Step 5 — Configure the Strategy Tester

Field	Recommended value
Expert	V1N1_LONNY_OPT (the working copy)
Symbol	The symbol to optimize, using your broker's exact name (EURUSD.m, XAUUSD.raw, etc.)
Period	M15 — the EA works on M15/M30/H1 internally through <code>OnTimer()</code> , regardless of the chart, but keeping it consistent makes results comparable
Modelling	1 minute OHLC to sweep the 540 combinations; then re-validate the best sets with Every tick based on real ticks
Date range	At least 2 years, so the EA produces enough trades. Personally, I use 5 years — it spans different market cycles and reduces the chance that the winning set is only good in one particular regime
Forward	1/4 or Custom — reserves a final slice for out-of-sample validation
Delay	Random delay (the EA works with pending orders, so latency matters)
Deposit / Leverage	The same as your live account
Optimization	Slow complete algorithm — with 540 passes it is perfectly feasible and avoids the local minima of the genetic algorithm
Optimization criterion	Custom max (the value returned by <code>OnTester()</code>)

About the custom criterion: the EA implements `OnTester()` with a composite function that weighs profit, profit factor, drawdown, number of trades, Sharpe ratio and recovery factor. Selecting Custom max is what makes the Tester rank the passes by that function. Any other criterion — "Balance max", for instance — throws that protection away and tends to reward sets with few trades and high drawdown.

Step 6 — Configure the inputs (the critical part)

In the Inputs tab:

6.1 — Leave the Symbol List EMPTY

Symbol List (empty = current symbol) → (clear the whole field)

This is the point that causes the most confusion. The EA behaves as follows:

- List filled in → it trades every symbol on the list, ignoring which symbol is selected in the Tester. The result would be for the whole portfolio, not for the symbol you want to optimize.
- List empty → `InitializeSymbols()` falls back to the current symbol, which is exactly the symbol selected in the Tester.

So: the symbol goes in the Tester, and the Symbol List field stays blank.

6.2 — Set the server time settings

Server GMT Offset	→ your broker's offset, in hours
Server DST Convention	→ your broker's daylight saving convention

The strategy is session-based (Asian range + London open breakout). With the wrong offset the time windows are shifted and the optimization result is worthless.

6.3 — Neutralize what is not being optimized

Risk Mode	→ Fixed Lot
Risk Size	→ the symbol's minimum lot (e.g. 0.01)
Daily Profit Limit	→ 0 (disabled)
Daily Loss Limit	→ 0 (disabled)

With a fixed lot, the ranking measures signal quality rather than the effect of compounding — two sets with the same signal stop being ranked merely by the order in which the trades happened. And with the daily limits off, no day is cut short, which avoids masking the effect of the parameters. Once you have picked a set, switch percentage risk and the daily limits back on for the final test.

6.4 — Check only the 5 optimization parameters

Tick the checkbox only on the 5 rows of the OPTIMIZATION SETTINGS group. Since they are enumerations, the Tester already lists every valid value — just tick them and leave the full range. All other inputs stay unchecked (fixed).

Step 7 — Run and analyze

In the Optimization Results tab, sort by the custom criterion column (`Result`).

What to look for:

- Prefer plateaus over isolated peaks. If the best set is excellent while its neighbours (same values with one parameter moved to the adjacent step) are poor, that is overfitting — the market will not reproduce that exact point. Look for a region where several nearby sets perform similarly, and take one from the middle.
- Check the Forward tab. A set that wins in-sample and collapses in the forward period was memorized, not fitted.
- Discard sets with few trades. Below roughly 100 trades over the period the result is statistically fragile, however good it looks.
- Run a single backtest (not an optimization) with the chosen set, now using `Every tick based on real ticks`, to confirm the behaviour under realistic spread and execution.

Step 8 — Write the result back into the table

Go back to the original EA and replace the numbers in that symbol's row with the winning values.

Watch the PSAR scale. The optimization input uses the enumeration multiplied by 100, while the table stores the decimal value:

Optimizer (ENUM_STEP_SAR)	Table (stepSAR)
STEP_SAR_002 = 2	0.02
STEP_SAR_009 = 9	0.09
STEP_SAR_016 = 16	0.16

It is the only parameter that needs conversion. The other four go in as-is (including PERIOD_H1_D = 0, which means "dynamic period").

Example — if the optimizer returned ADR = 24, H1 = 14, SAR = STEP_SAR_009, MACD = 21 and Asian = 8:

```
SetSymbolParams(i++, "EURUSD", 24, 14, 0.09, 21, 8);
```

Step 9 — Repeat, then validate the portfolio

Go back to Step 3 and do the next symbol, always one at a time.

Once the list is done, run a final test with the original EA, now with the Symbol List filled in with every symbol and percentage risk re-enabled. That measures the portfolio: correlation between pairs, simultaneous margin usage and aggregate drawdown — numbers no individual optimization can show.

Quick summary

1. Copy the EA to an _OPT version
2. Uncomment the OPTIMIZATION SETTINGS block
3. Replace the 5 numbers in the symbol's row with the OPT_PARAM_* variables
4. Compile
5. Tester: target symbol + Custom max + slow complete algorithm + 5 years of history
6. Symbol List empty + correct server time settings + fixed lot
7. Analyze looking for a plateau, not a peak
8. Write the values into the original EA's table (PSAR ÷ 100)
9. Next symbol

Final reminders

- Re-optimize periodically (every 6–12 months), but do not overdo it: changing parameters after every rough patch is chasing noise.

- Always optimize on the broker where the EA will run. Spread, server time and history differ from one to another, and the default table values were produced in a different environment.
- Before going live, run the final set on a demo account for a few weeks.