

# Parameters in StelvioUI.ini

All user parameters with their default values.

## General parameters:

- `inputFileName` = `problems.txt`
- `outputFileName` = `problems_out.txt`
- `startAtStrategyNr` = 1. This can be set to a high value in case the first x strategies should be skipped. Also applies to histogram mode.
- `histogramMode` = `false`. Instead of playing strategies, only count them and display them in a histogram. Default is `false`. Starts counting at the strategy defined by `startAtStrategyNr`.
- `printStrategies` = `false`. Print all strategies into the output file. This can easily fill your disk in some cases, as strategies can come in the tens of millions. Nice.
- `maxSolutionsPerCook` = 2. Number of cook-solutions to add to the output file per cook-strategy. Maximum is 4.
- `stopAfterXCooks` = 1. Abort the solving process after finding this number of cook strategies.
- `pgnOutput` = `false`. Write solutions/cooks in pgn format. Actually, the output moves are always fully qualified (like in the non-pgn case), which is sufficient for use in tools like lichess.org.

## Cache parameters:

- `positionCacheMaxOverallExponent` = -1. Technical value to limit the main cache size which is usually guessed by the given RAM available. It can be useful, albeit rarely, to downsize the cache in the special case when the rest of the needed artifacts require a lot of memory. One example would be a problem with many promoted rooks, as there are then a huge number of permutations in memory, which rook is what piece initially. Values below 25 are ignored.  $2^{\text{positionCacheMaxOverallExponent}}$  is used.
- `positionCacheSplitExponent` = 2. How much should we split up the huge cache array into parts. Values 2-10. This has a performance impact, but has to be analyzed still to know what is best.
- `positionCacheMaxChainLength` = 500. How long should we look for an empty slot in the cache / try to find our position in the cache. Nobody knows what is best here, but value should probably be  $> 100$ . This also has a performance impact
- `positionCacheEvictionType` = `byHalfMove`. There are two caches to choose from, which differ among other things by eviction type. As one of them is quite new, there is not enough data to decide which one should be used, so both are available. Valid values are `byHalfMove` and `full`.
- `positionCacheFullEvictionThreshold` = 0.85. The threshold in cache size that triggers eviction. Between 0 and 1. A value  $> 1$  means never. This also has a performance impact.
- `positionCacheSizeAdjustment` = 0. The secondary cache can be made slightly larger or smaller. In case other artifacts need a lot of memory, this can be set to -1, decreasing this cache's size. The value 1 on the other hand increases the secondary cache to the maximum, but if you are unlucky, Stelvio will blow up with an out of memory error. Possible values are -1, 0, 1. This only has an effect in case `positionCacheEvictionType` = `byHalfMove`.