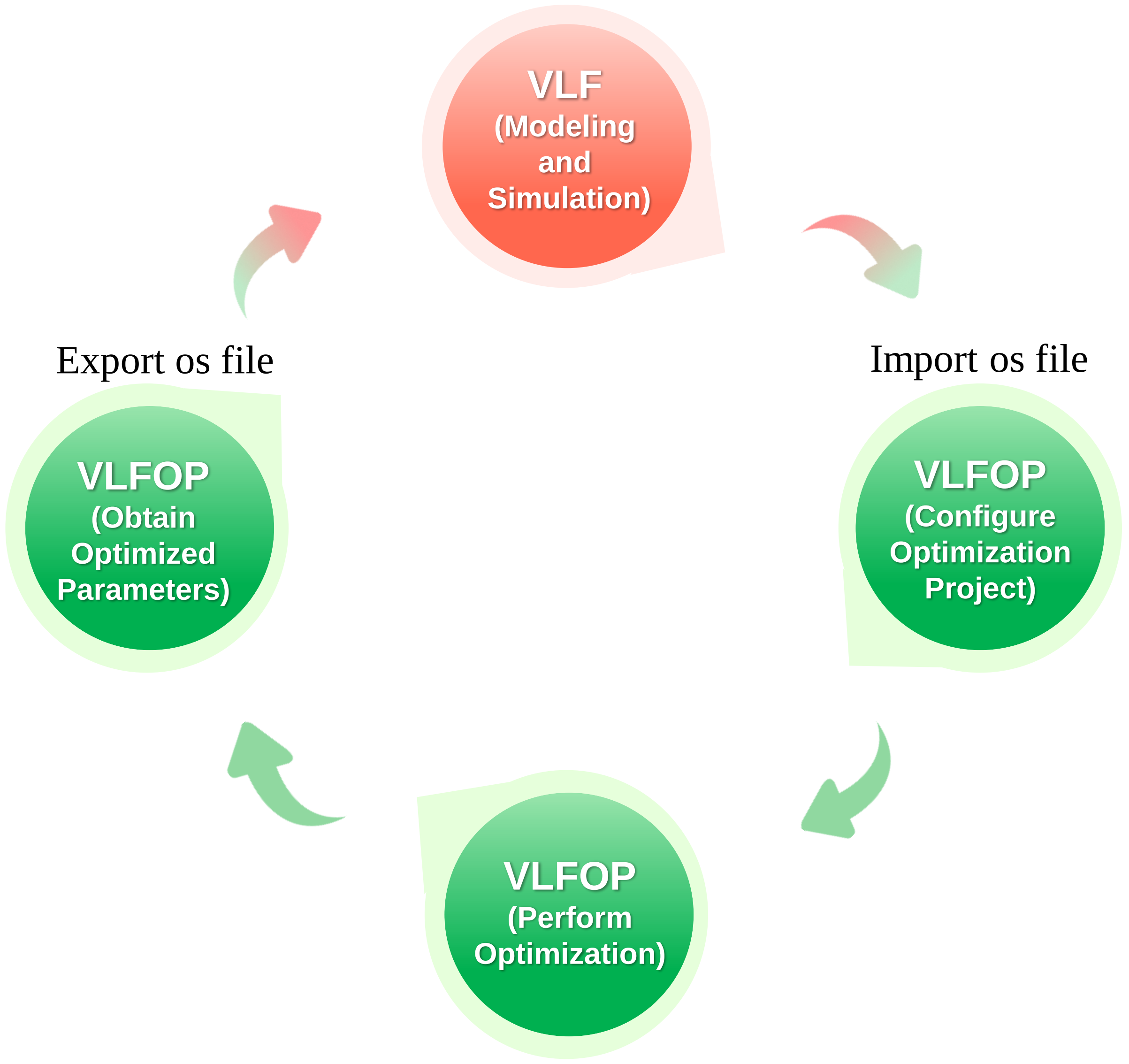
Introduction to the VirtualLab Fusion Optimization Package

Use Case :Introduction to the VirtualLab Fusion Optimization Package

Abstract

VirtualLab Fusion Optimization Package is a dedicated optimization tool designed to complement the VirtualLab Fusion. As a powerful optical system optimization software, VLFOP offers a variety of advanced optimization algorithms to assist users in optimizing complex optical systems, enhancing system performance, and efficiency. This use case explains how to use VLFOP, including project import and export, different configurations, and algorithms.

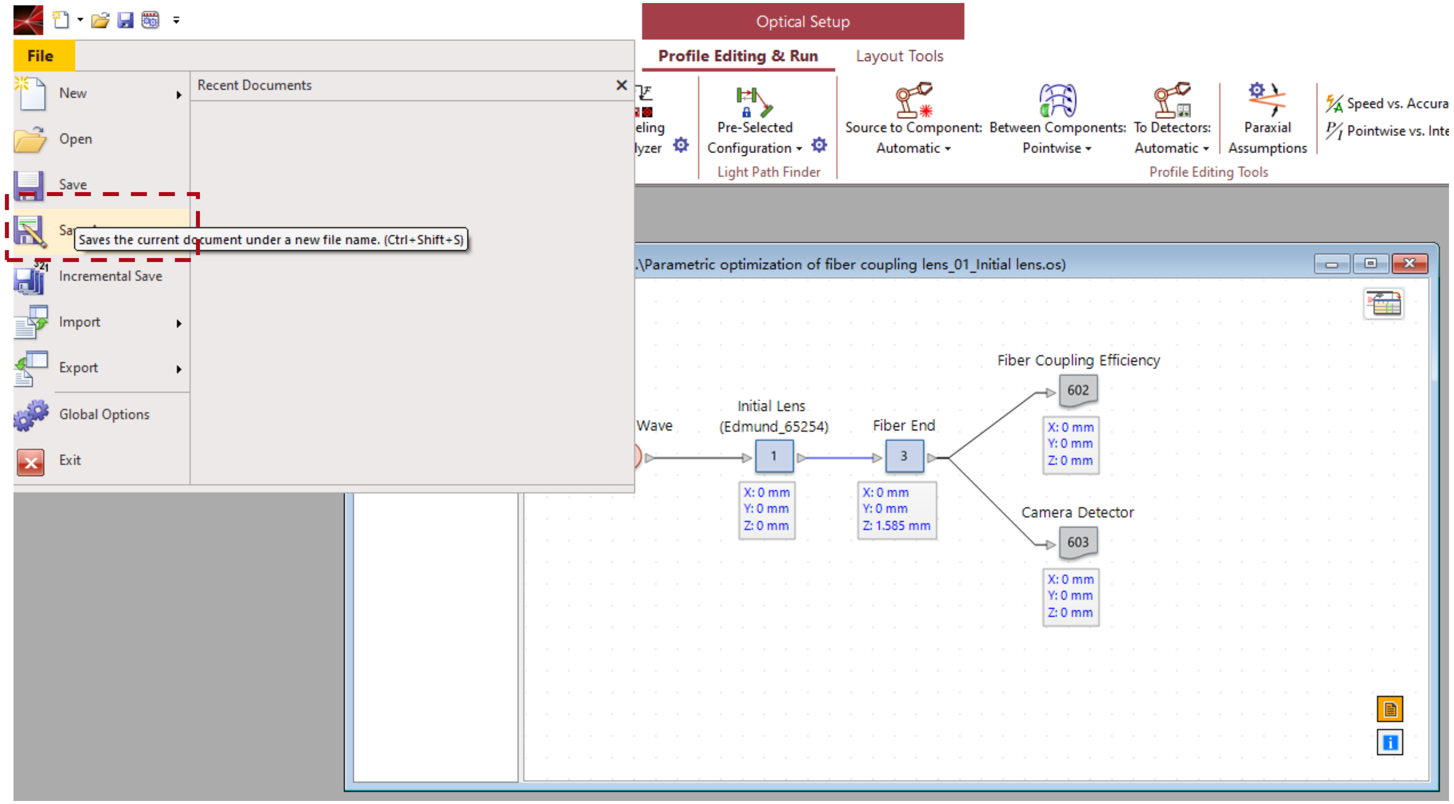
Workflow



VirtualLab Optimization Package is an optimization tool for VirtualLab Fusion, designed to optimize various optical systems within VLF. It offers robust optimization capabilities and can create optimization projects based on os files from VLF. Once an optimization project is created, it can be configured and executed to obtain optimization results. These results can then be exported and imported back into VLF, yielding an optimized optical system.

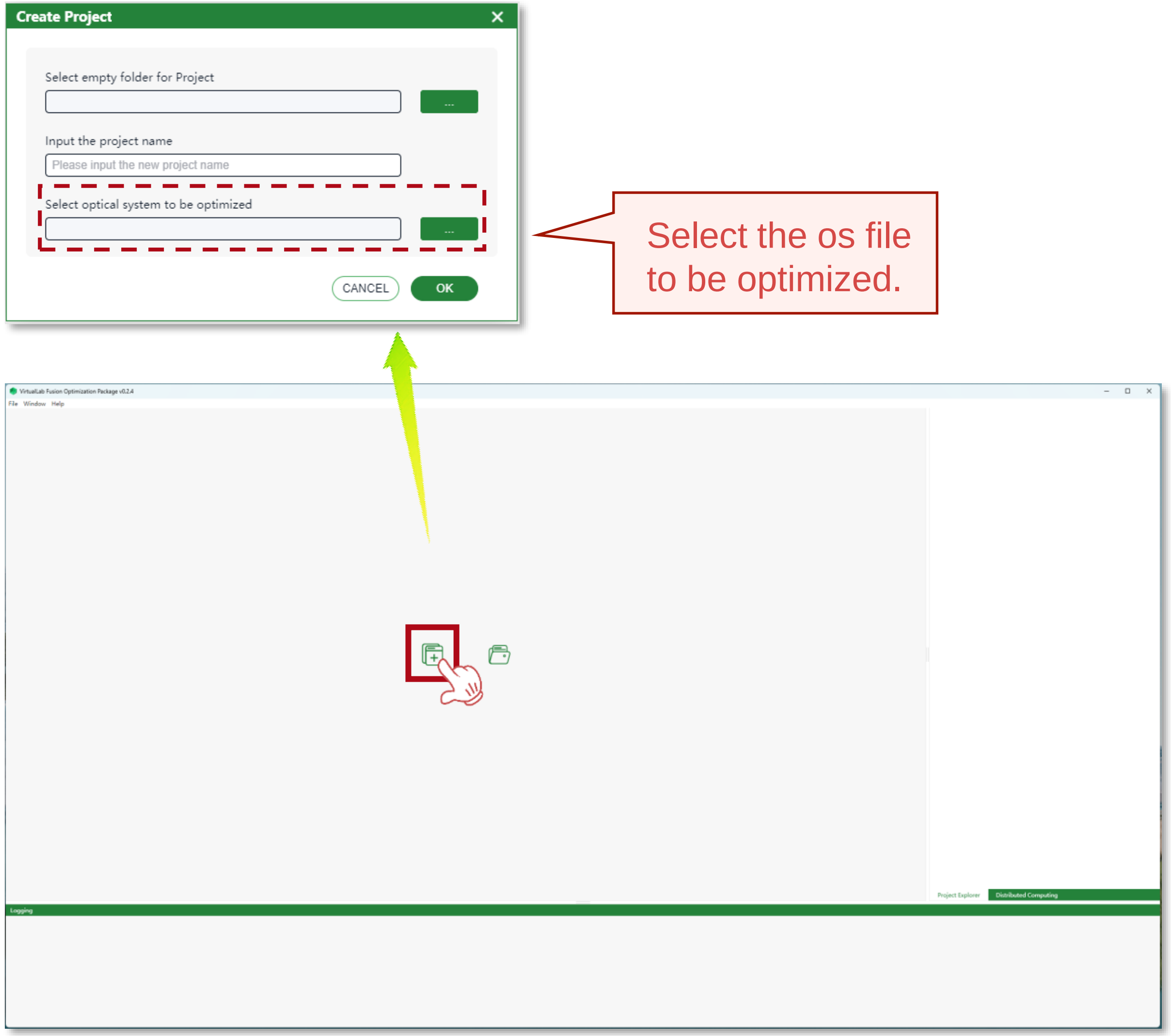
Modeling in VLF

Click "Save As" to save the current optical system as an os file.



VLFOP is an optimization software used in conjunction with VLF. Users need to complete the modeling of the optical system and the setup of the detectors in VLF, and then export the system as an os file.

Creating a New VLFOP Project



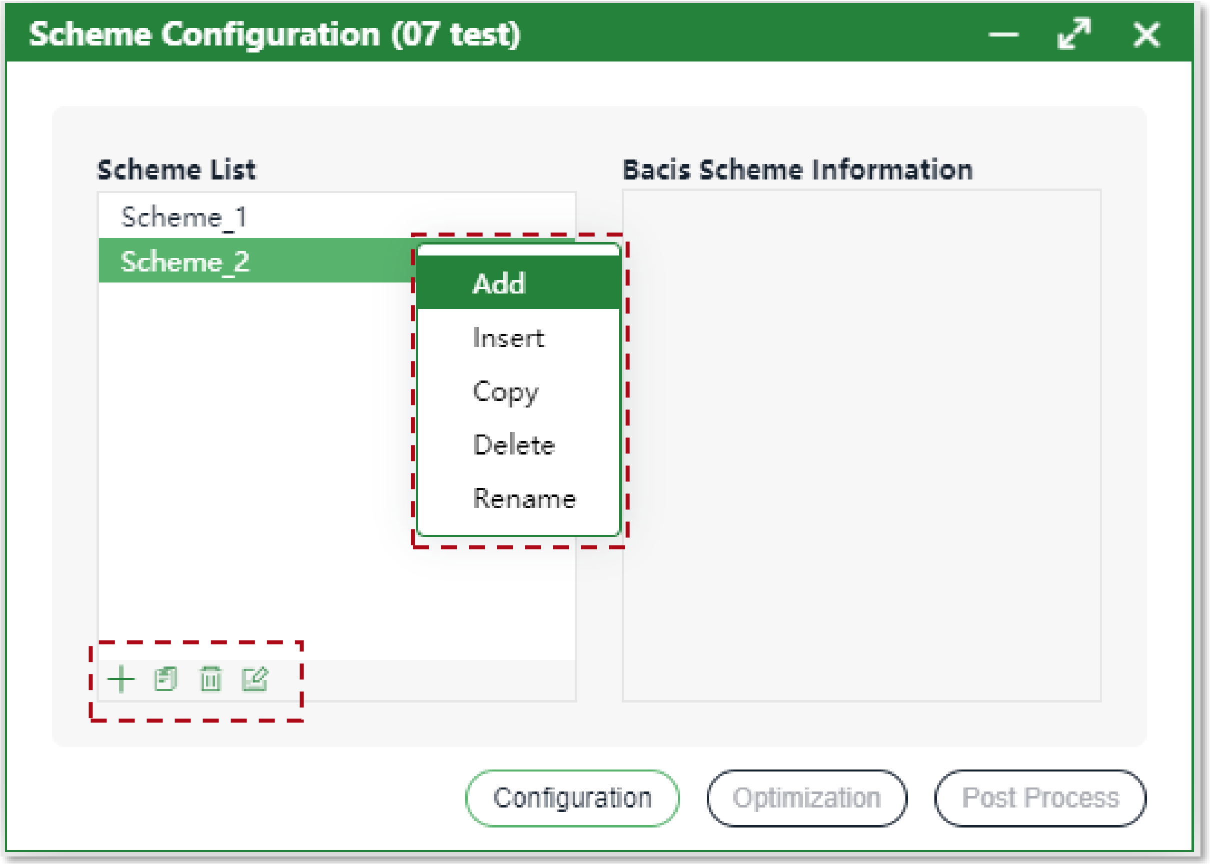
There are three ways to create a new project.

- After opening the software, click  on the start screen to open the "Create Project" dialog box.
- The file menu> New Project
- The shortcut “Ctrl+N”

Scheme Configuration

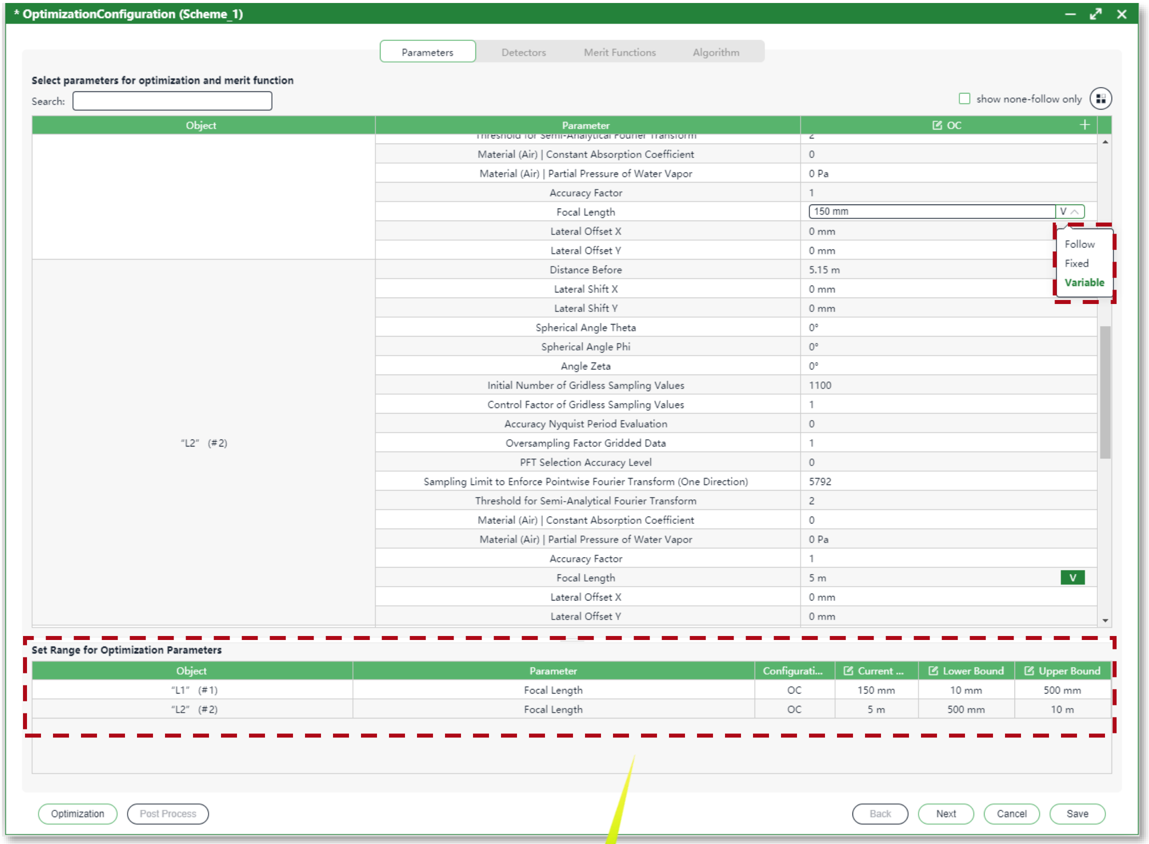
There are two ways to manage schemes.

- Manage through the right-click menu.
- Manage through the toolbar at the bottom of the scheme list.



VLFOP supports using different optimization schemes within a single optimization project. When creating a new project or opening an existing one, the Scheme Configuration Dialog will appear. In this dialog, you can create and save multiple schemes and manage them.

Parameter Selection



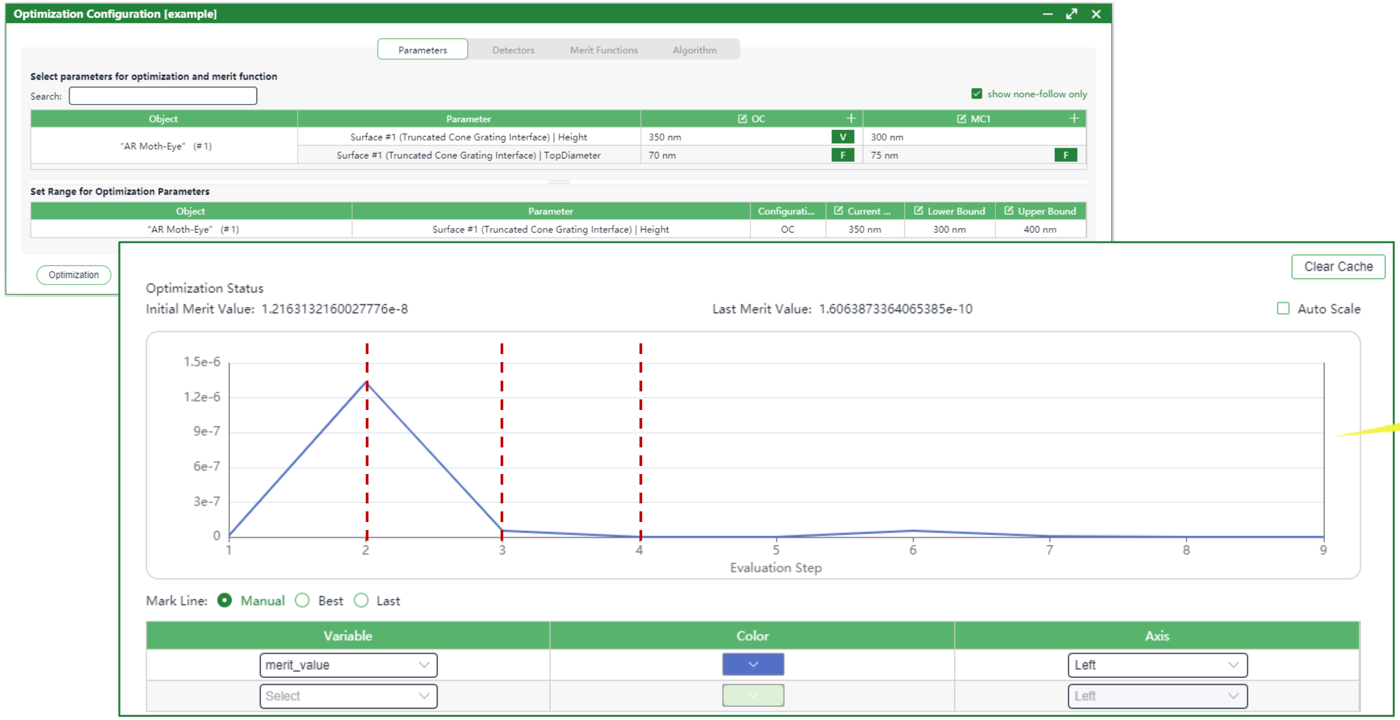
The parameters set as variables need to have their bounds specified in the table below.

Through the parameter list, users can select parameters for optimization or designate parameters to remain fixed during optimization. At least one parameter must be selected as a variable

Parameter Type Selection

- Follow: This parameter changes in accordance with the preceding configuration (where the parameters of OC follow the configuration parameters in the OS file)
- Fixed: This parameter remains constant throughout the entire optimization process.
- Variable: This parameter can be adjusted or modified in an optimization problem.

Parameter Selection



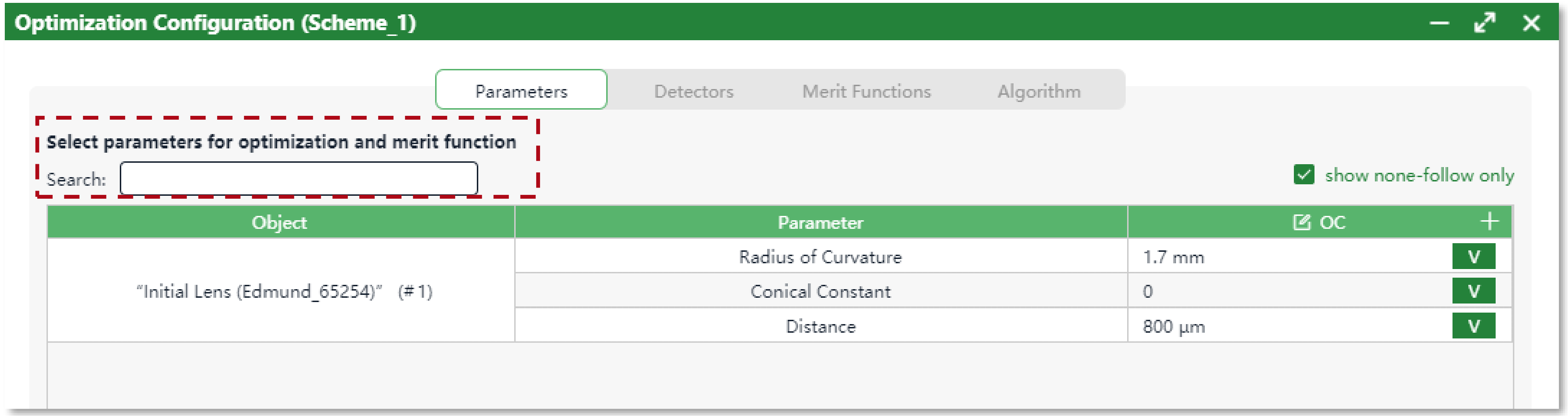
Evaluation Step	Parameter	OC		MC1	
2	Height	367.5nm	V	367.5nm	FO
	Top Diameter	70nm	FIX	75nm	FIX
3	Height	332.5nm	V	332.5nm	FO
	Top Diameter	70nm	FIX	75nm	FIX
4	Height	345.63nm	V	345.63nm	FO
	Top Diameter	70nm	FIX	75nm	FIX
…					

Examples of parameter types are as follows:

- Variable: OC-Height continuously changes during the optimization process through the optimization algorithm.
- Fixed: OC-Top Diameter and MC1-Top Diameter remain constant during the optimization process.
- Follow: MC1-Height remains consistent with OC-Height during the optimization process.

Parameter Selection

show none-follow only to filter out the unconcerned parameters

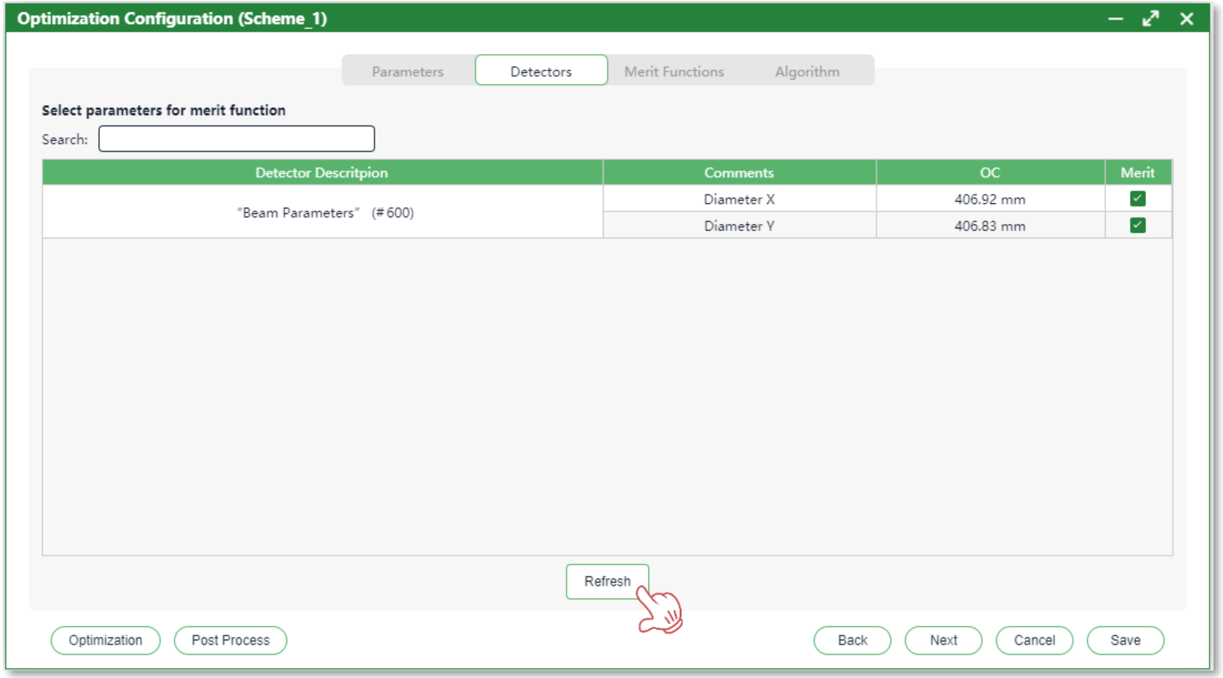


Some features for a better overview

- Keyword search
- It can be chosen that only the varied parameters are shown

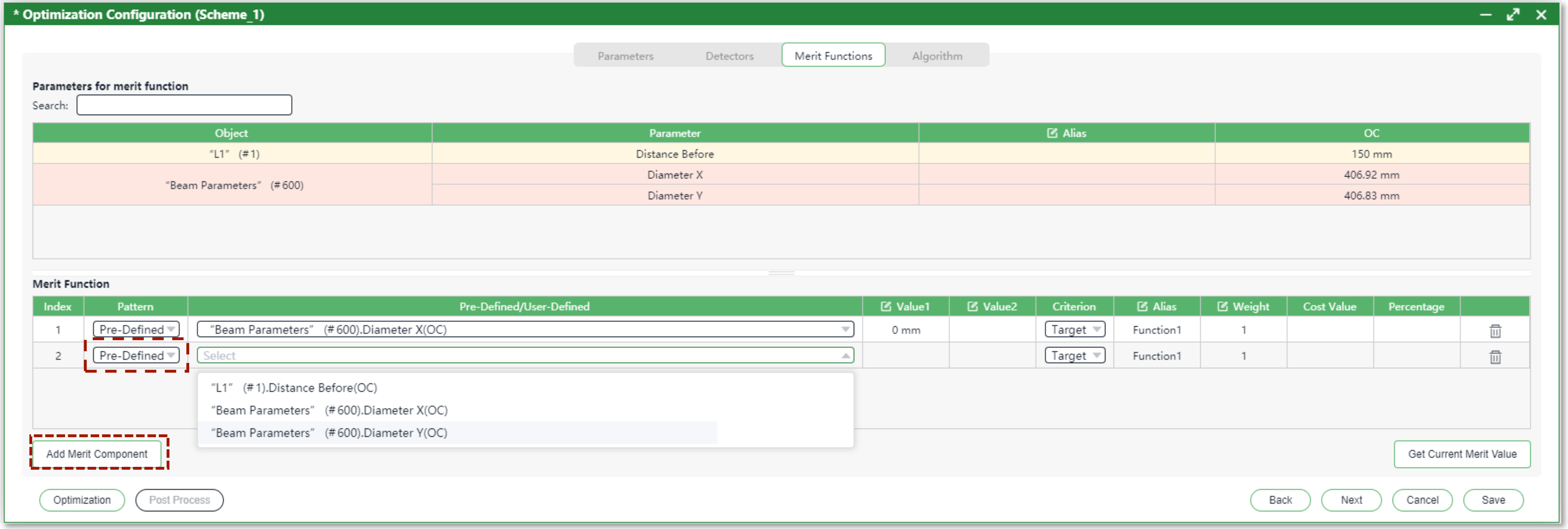
Detector Selection

Click the "Refresh" button to see the values of all detection results in the current configuration.



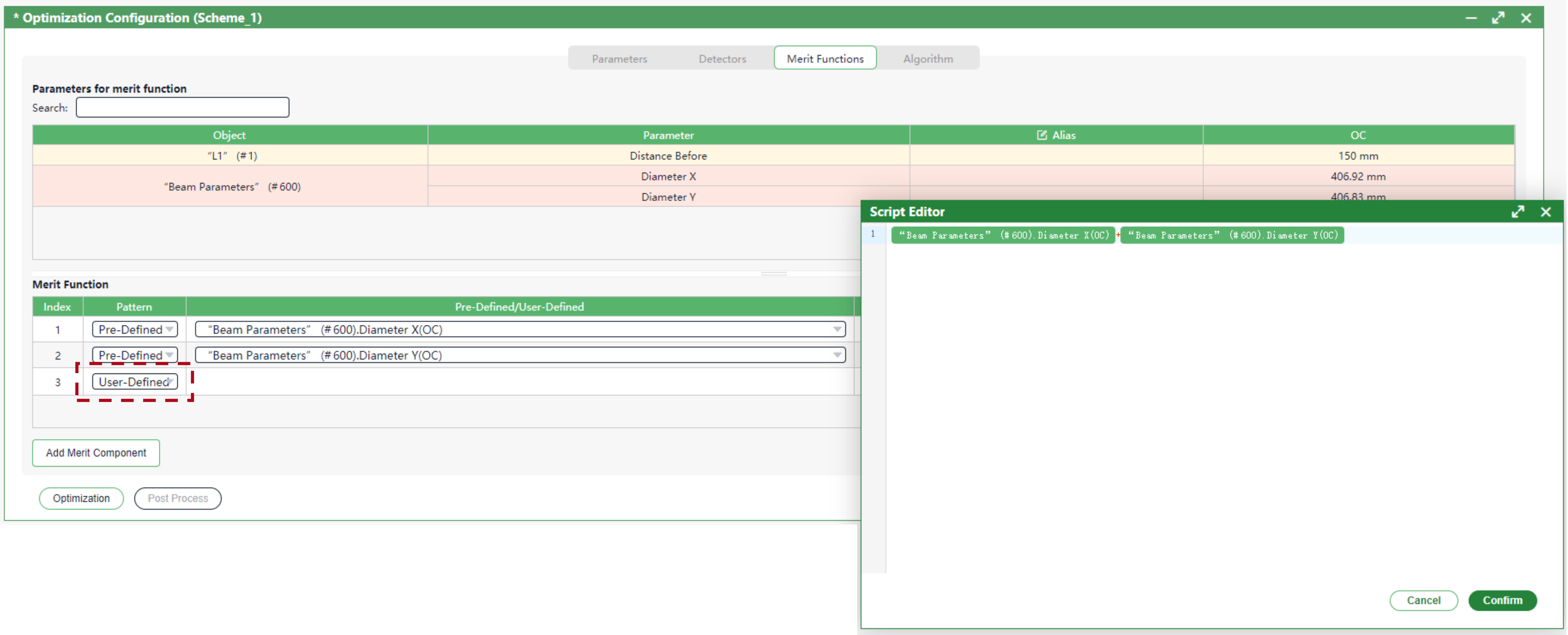
This page will display the detectors and analyzers configured in VLF. Only the items checked in the "merit" column are eligible to be set as components of the merit function.

Merit Function Configuration



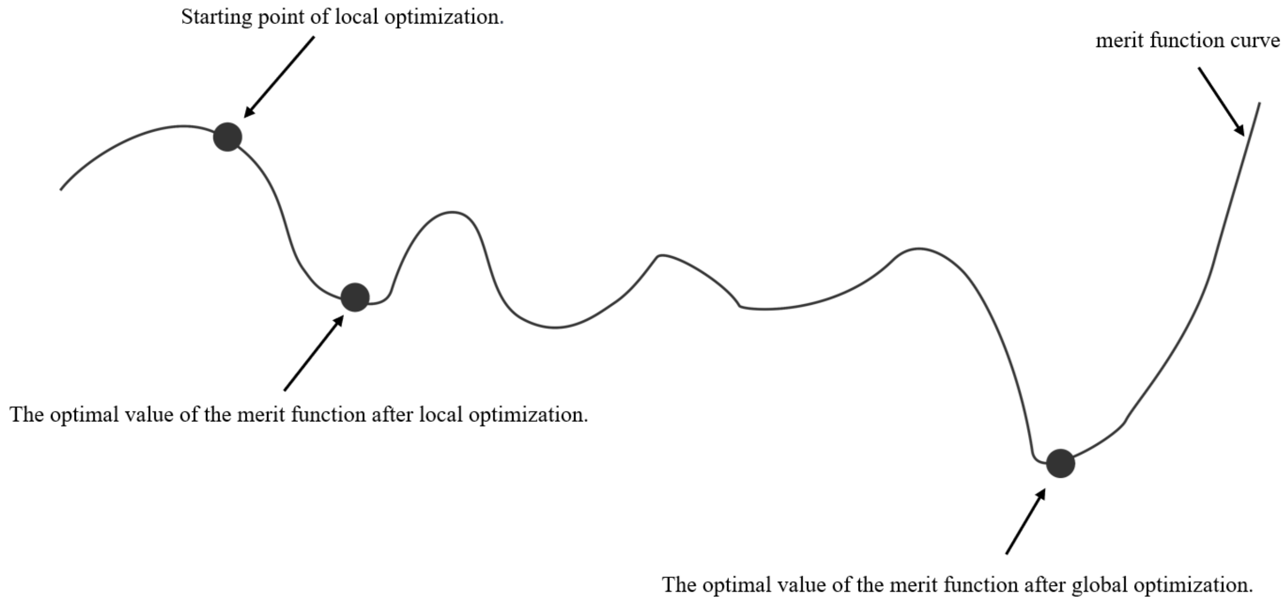
On this page, the user can set the Merit Function Component and the specify Criterion type. By clicking the "Add Merit Component" button, you can add a new merit function component. There are two modes for merit function components, and in the “Pre-Defined” mode, users can set any parameter from the “Parameters for merit function" table as a merit function component.

Merit Function Configuration



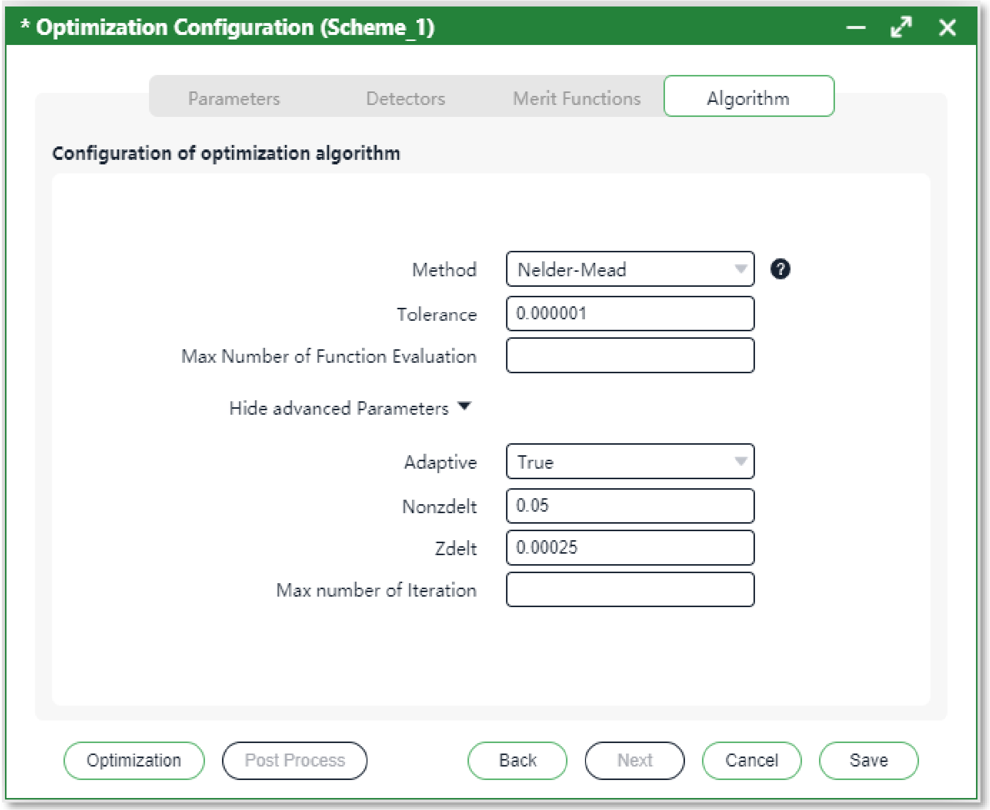
Another pattern of merit function component is "user-defined", Choosing User-Defined Pattern can support performing mathematical operations on parameters from the table above to create a custom merit function Component.

Local & Global Optimization



The two main strategies in optimization problems are local optimization and global optimization. Local optimization typically converges quickly but can easily get trapped in local optima. On the other hand, global optimization aims to find the global optimum within the entire search space, avoiding the pitfalls of local optima. Currently, we offer a local optimization algorithm (Nelder-Mead) and a global optimization algorithm (Differential Evolution).

Algorithm Selection: Nelder-Mead

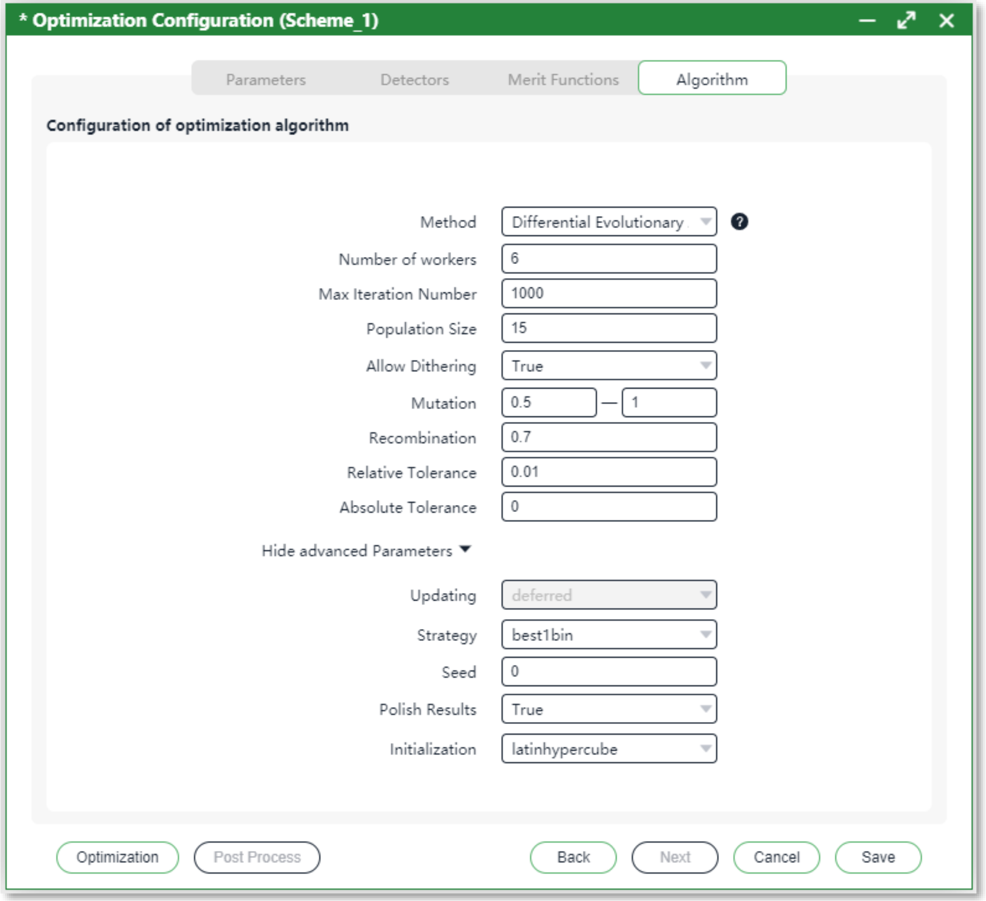


The Nelder-Mead algorithm is a common local optimization algorithm, particularly well-suited for unconstrained nonlinear optimization problems. It works by constructing a simplex and iteratively adjusting the positions of the vertices in the parameter space to find the optimal solution.

Item	Description
Adaptive	Adapt algorithm parameters to dimensionality of problem. When "adaptive" is selected, both the expansion and contraction parameters are determined by formulas. Useful for high-dimensional minimization
Nonzdelt	nonzdelt is a parameter used to adjust non-zero dimensional values when generating the initial simplex. When a certain dimension value of the original point is non-zero, the value of that dimension in the new point is adjusted to (1 + nonzdelt)
Zdelt	zdelt is a parameter used to adjust zero-dimensional values when generating the initial simplex. When a certain dimension value of the original point is zero, the value of that dimension in the new point is set to zdelt to introduce a non-zero offset. This helps increase the diversity of the initial simplex, aiding the algorithm in exploring the optimization space more effectively.
Tolerance	Absolute error in xopt and func between iterations that is acceptable for convergence. Both convergence criteria must be satisfied simultaneously for the optimization to stop.
Maxfev	Maximum allowed number of function evaluations. Will default to N*200, where N is the number of variables. If both Max Function Evaluation Number and Max Iteration Number are set, minimization will stop at the first reached.
Maxiter	Maximum allowed number of iteration. Will default to N*200, where N is the number of variables. If both Max Function Evaluation Number and Max Iteration Number are set, minimization will stop at the first reached.

Regarding the meaning of more parameters in this algorithm, please refer to: Nelder-Mead Algorithm.

Algorithm Selection: Global Algorithm

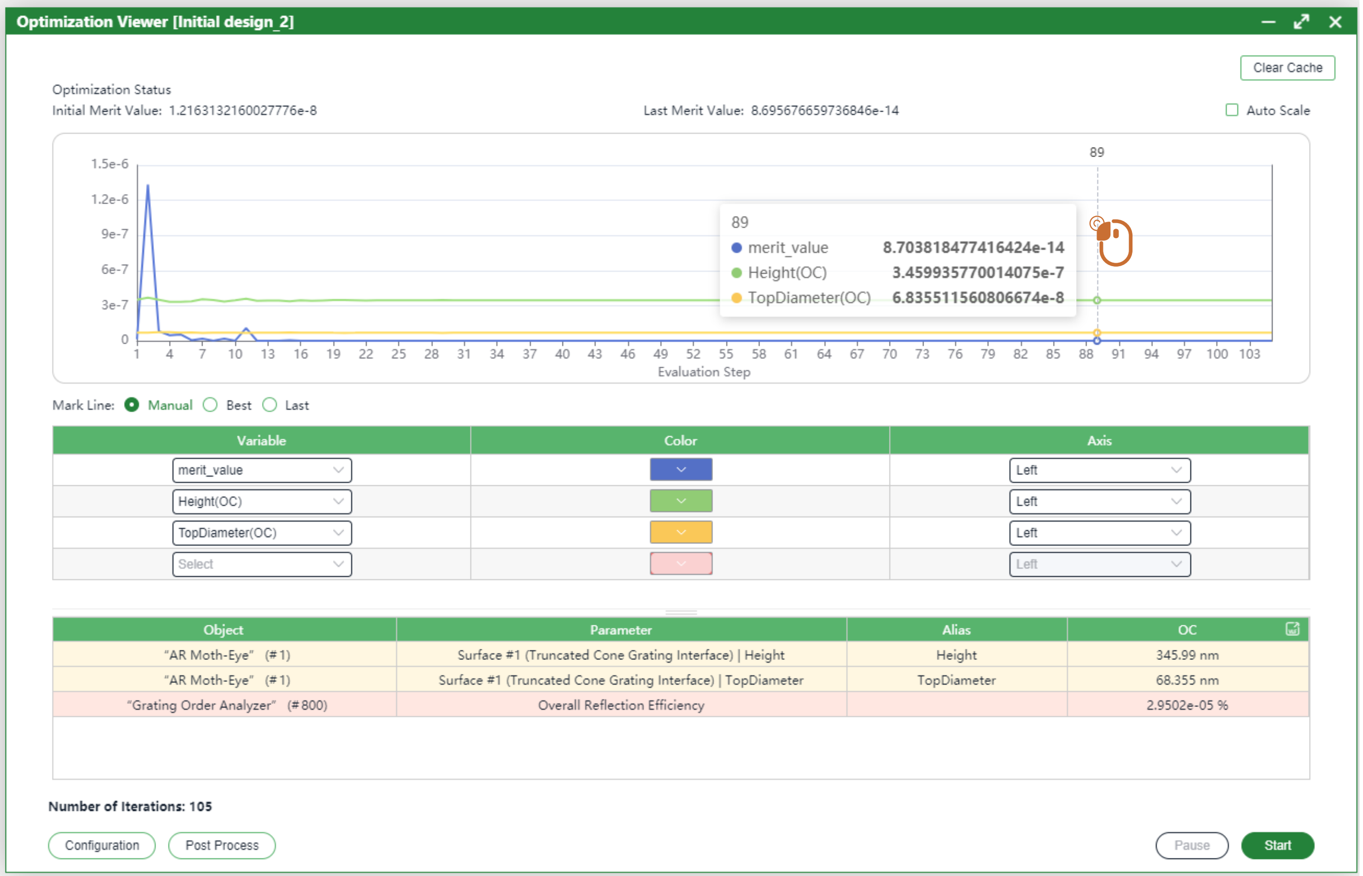


Currently, we have only one global optimization algorithm, which is the Differential Evolution (DE) algorithm. This algorithm has the advantages of strong global search capability, high adaptability, simplicity, and fast convergence.

Item	Description
Number of workers	If Number or workers is an integar value the population is subdivided into Number or workers sections and evaluated in parallel. Supply 0 to use all available CPU cores. Alternatively supply a map-like callable
Max Iteration Number	The maximum number of generations over which the entire population is evolved. The maximum number of function evaluations is: (maxiter + 1) * popsize * (N - N_equal). N is the number of parameters, and N_equal is the number of parameters whose bounds are equal.
Population Size	A multiplier for setting the total population size. The population has population size * (N - N_equal) individuals. When using Initialization='sobol' the population size is calculated as the next power of 2 after population size * (N - N_equal).
Allow Dithering	Dithering is an option. Dithering randomly changes the mutation constant on a generation by generation basis. The mutation constant for that generation is taken from U[min, max). Dithering can help speed convergence significantly. Increasing the mutation constant increases the search radius, but will slow down convergence.
Mutation	The mutation constant, which is also known as differential weight, being denoted by F. If specified as a float it should be in the range [0, 2]. If mutation constant specified as a tuple (min, max) dithering is employed.
Recombination	The recombination constant, should be in the range [0, 1],which is also known as the crossover probability, being denoted by CR. Increasing this value allows a larger number of mutants to progress into the next generation, but at the risk of population stability.
Tolerance	the solving stops when np.std(pop)<= atol + tol*np.abs(np.mean(population_energies)), where and atol and tol are the absolute and relative tolerance respectively.

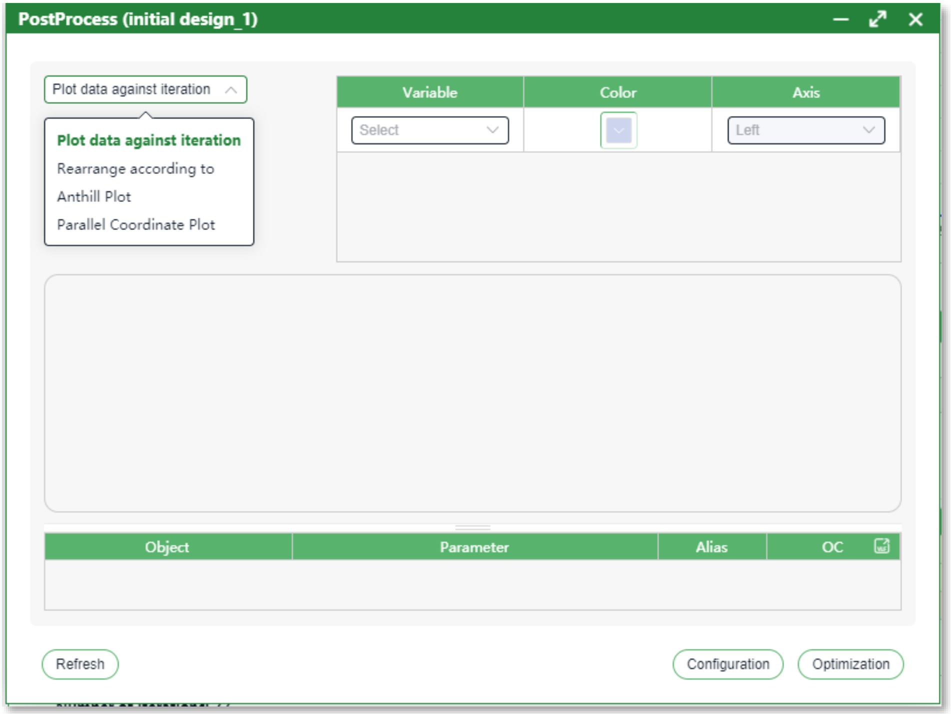
Regarding the meaning of more parameters in this algorithm, please refer to: Differential Evolution Algorithm.

Optimization Result



- When the optimization starts, the graph at the top will, by default, plot the current merit value versus the evaluation steps.
- Below the graph, users can choose which variables are displayed in the graph through the table.
- Clicking on the curve in the graph will display the values of the merit function components and variables at the current evaluation count.

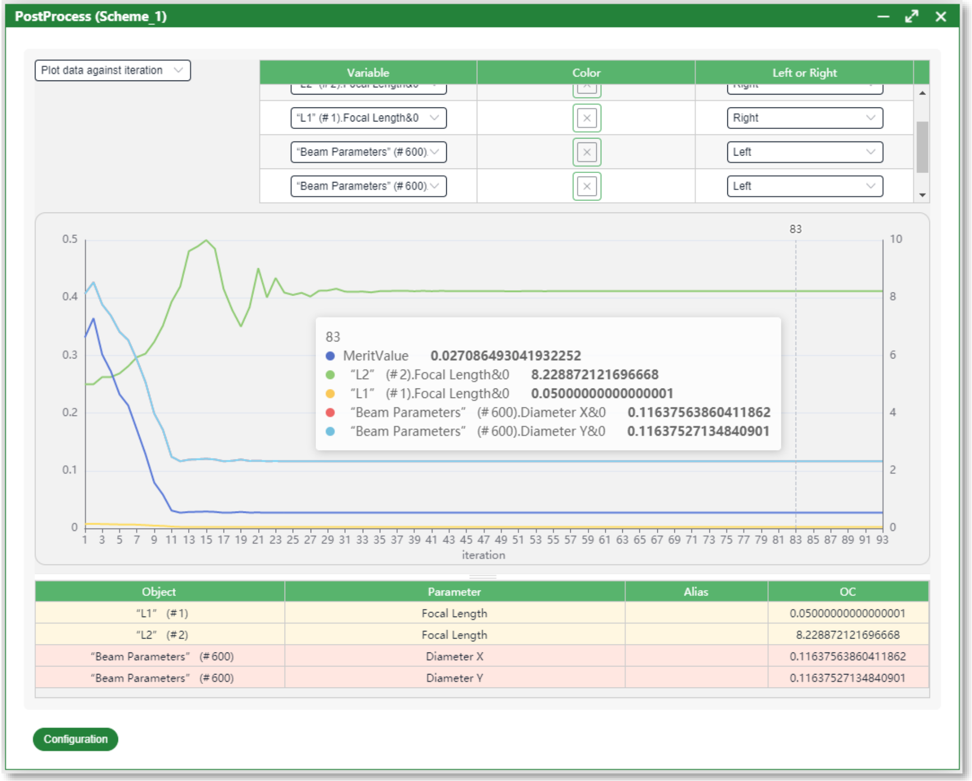
Post Process



The VirtualLab Optimization postprocessing tool has serval modes. Each mode incorporates a set of plots.The different postprocessing modes are:

- **Plot Data Against Evaluation Step.**
- **Rearrange According To Different Parameters.**
- **Anthill Plot.**
- **Parallel Coordinate Plot.**

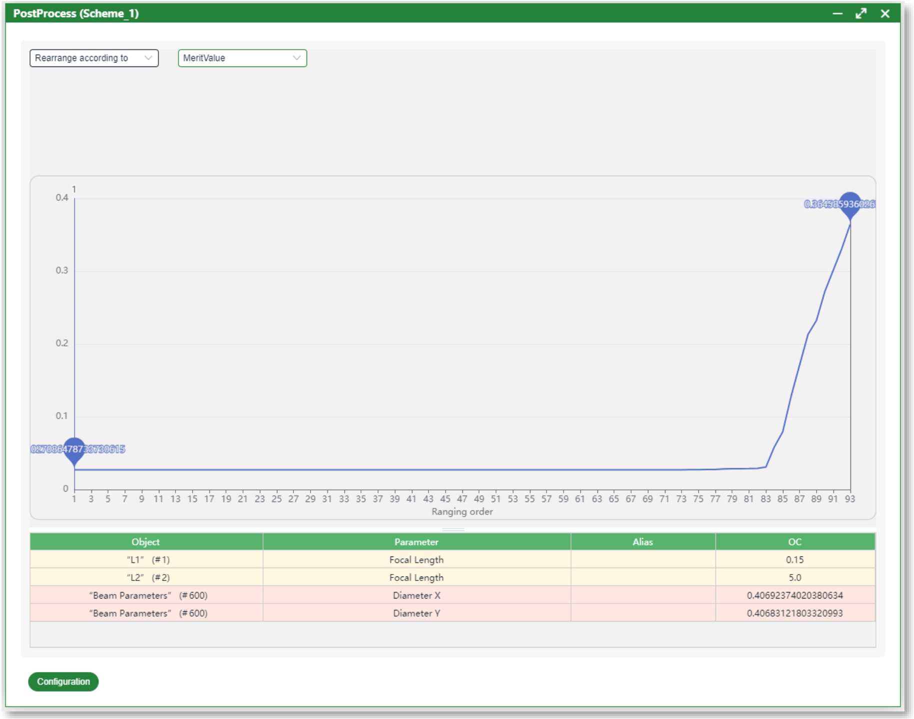
Post Process-Plot Data Against Evaluation Step.



"Plot data against iteration" will plot the changes in parameters over the evaluation step.

- User can choose which variables and merit functions to display on the plot.
- The table below shows the values of all merit functions and variables at this step.
- This mode is identical to the Optimization Viewer.

Post Process-Rearrange According To.



"**Rearrange according to** "mode is commonly used to filter out the minimum parameter during the optimization process.

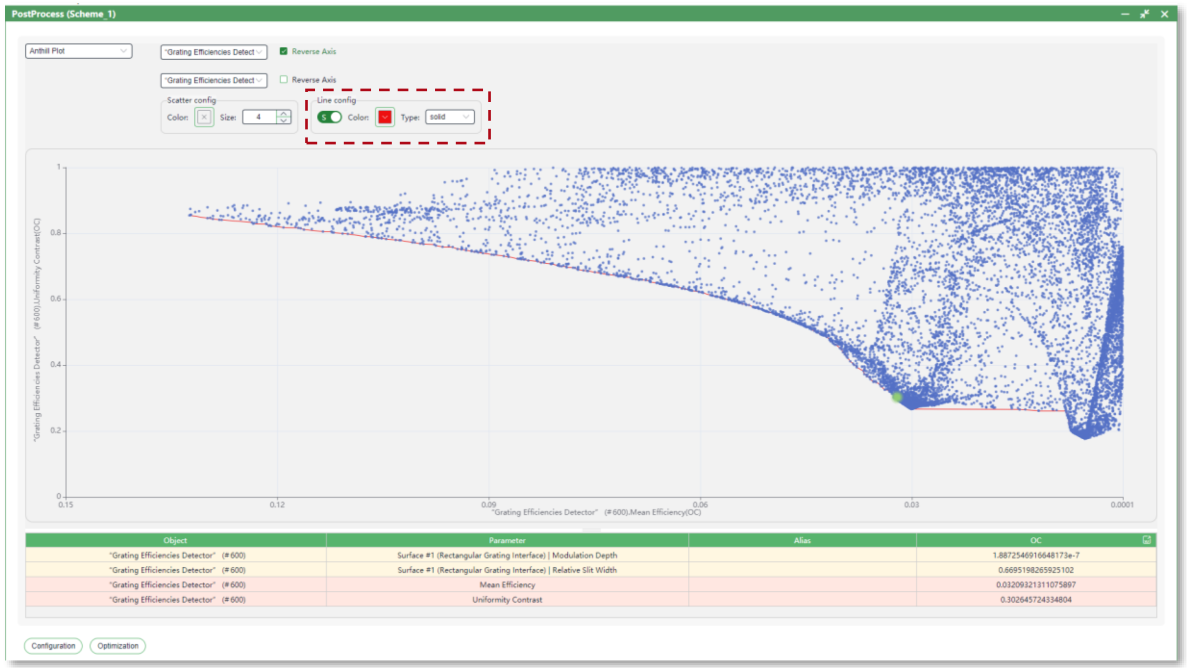
- The selected parameters will be rearranged in ascending order.
- The parameters of selected order will be displayed in a list below the plot.
- Above the plot, there will be an indication of which evaluation step in the optimization process corresponds to the selected order.

Post Process-Anthill Plot.

The **"Antill Plot"** is a valuable tool in the field of optimization, especially when dealing with complex system.

- Visualizing Search Paths: The anthill plot can show the paths and behavior of the optimization algorithm in the search space.
- Identifying Local Minima: By observing the anthill plot, one can identify whether the algorithm has fallen into local minima and the distribution of these local minima.
- Discovering Search Regions: The anthill plot helps to understand the exploration range of the algorithm in the search space, determining if there are areas that have not been sufficiently searched.

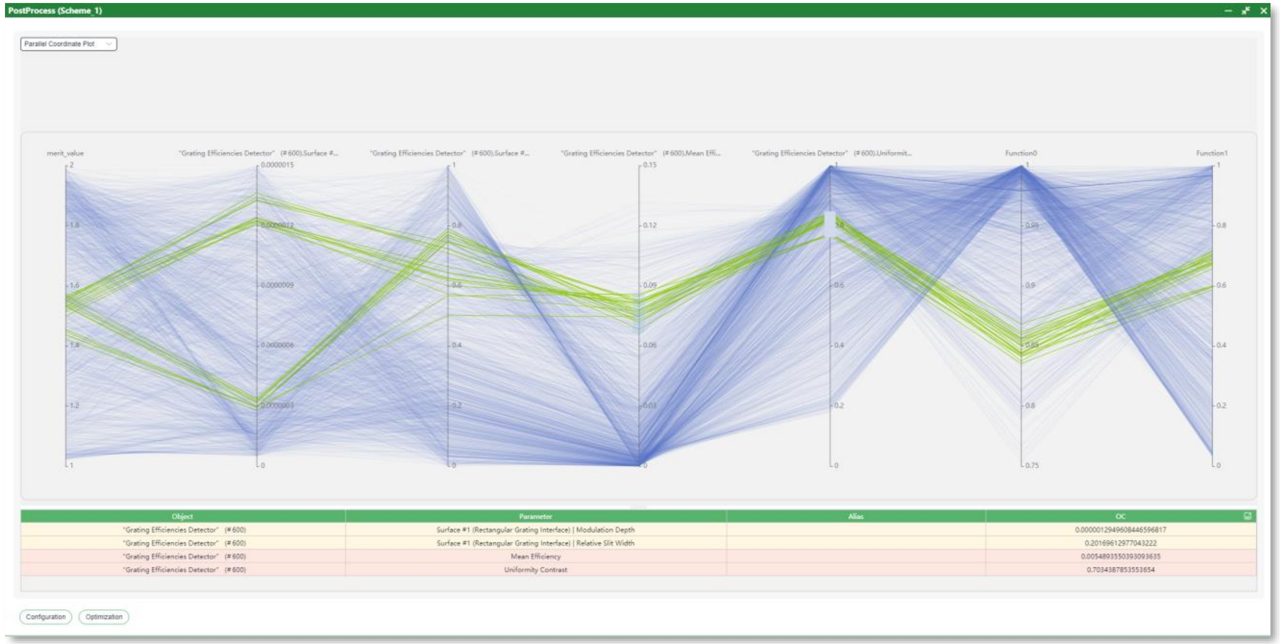
Post Process-Anthill Plot.



Switch the toggle to "show" to display the Pareto front in the graph.

When both the x-axis and y-axis represent components of the objective function, user can use the feature to display the Pareto front. It helps illustrate the trade-offs and compromises in multi-objective optimization.

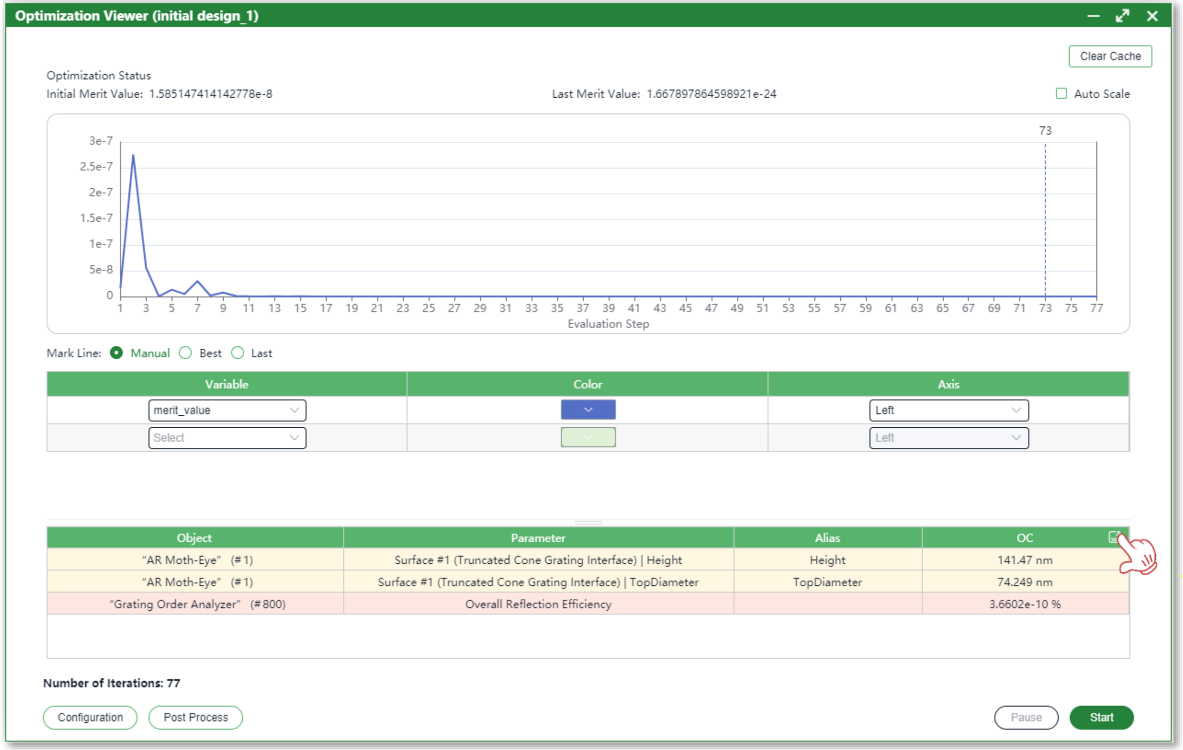
Post Process-Parallel Coordinate Plot.



“Process-Parallel Coordinate Plot” is a commonly used method for visualizing high-dimensional multivariate data.

- Axis Representation of Variables
- Trend Observation: By observing the shape of the line segments, trends in the distribution of data points across different variables can be identified.
- Pattern Recognition: Parallel coordinate plots assist users in recognizing patterns and correlations within the dataset.
- User can select and highlight specific data line for more in-depth exploration

Export Optimized System as OS File



Click 'Export as OS file Button on the 'OC' column header in the **Optimization Result Table**

Document Information

title	Introduction to the VirtualLab Fusion Optimization Package
document code	OPT.0001
document version	1.0
required packages	Optimization Package
software version	2024.1 (Build 1.132)
category	Application Use Case
further reading	