Global Optimization Toolbox provides methods that search for global solutions to problems that contain multiple maxima or minima. It includes global search, multistart, pattern search, genetic algorithm, and simulated annealing solvers. You can use these solvers to solve optimization problems where the objective or constraint function is continuous, discontinuous, stochastic, does not possess derivatives, or includes simulations or black-box functions with undefined values for some parameter settings.

Genetic algorithm and pattern search solvers support algorithmic customization. You can create a custom genetic algorithm variant by modifying initial population and fitness scaling options or by defining parent selection, crossover, and mutation functions. You can customize pattern search by defining polling, searching, and other functions.

**Key Features**

- Interactive tools for defining and solving optimization problems and monitoring solution progress
- Global search and multistart solvers for finding single or multiple global optima
- Genetic algorithm solver that supports linear, nonlinear, and bound constraints
- Multiobjective genetic algorithm with Pareto-front identification, including linear and bound constraints
- Pattern search solver that supports linear, nonlinear, and bound constraints
- Simulated annealing tools that implement a random search method, with options for defining annealing process, temperature schedule, and acceptance criteria
- Parallel computing support in multistart, genetic algorithm, and pattern search solvers
- Custom data type support in genetic algorithm, multiobjective genetic algorithm, and simulated annealing solvers
Plot of different types of problems Global Optimization Toolbox can solve: nonsmooth (top), single global minimum nested inside multiple local minima (left), multiple local minima with no global minimum (right).
Plot of a nonsmooth objective function (bottom) that is not easily solved using traditional gradient-based optimization techniques. The Optimization Tool (middle) shows the solution found using pattern search in Global Optimization Toolbox. Iterative results for function value and mesh size are shown in the top figure.

Defining, Solving, and Assessing Optimization Problems

Global Optimization Toolbox provides functions that you can access from the command line and from the Optimization Tool graphical user interface (GUI) in Optimization Toolbox™. Both the command line and GUI let you:

- Select a solver and define an optimization problem
- Set and inspect optimization options
- Run optimization problems and visualize intermediate and final results
- Use Optimization Toolbox solvers to refine genetic algorithm, simulated annealing, and pattern search results
- Import and export optimization problems and results to your MATLAB® workspace
- Capture and reuse work performed in the GUI using MATLAB code generation

You can also customize the solvers by providing your own algorithm options and custom functions. Multistart and global search solvers are accessible only from the command line.
Visualization of Rastrigin’s function (right) that contains many local minima and one global minimum (0,0). The genetic algorithm helps you determine the best solution for functions with several local minima, while the Optimization Tool (left) provides access to all key components for defining your problem, including the algorithm options.

The toolbox includes a number of plotting functions for visualizing an optimization. These visualizations give you live feedback about optimization progress, enabling you to make decisions to modify some solver options or stop the solver. The toolbox provides custom plotting functions for both the genetic algorithm and pattern search algorithms. They include objective function value, constraint violation, score histogram, genealogy, mesh size, and function evaluations. You can show multiple plots together, open specific plots in a new window for closer examination, or add your own plotting functions.
Run-time visualizations (right) generated while the function is being optimized using genetic algorithm plot functions selected in the Optimization Tool (left).

Using the output function, you can write results to files, create your own stopping criteria, and write your own application-specific GUIs to run toolbox solvers. When working from the Optimization Tool, you can export the problem and algorithm options to the MATLAB workspace, save your work and reuse it in the GUI at a later time, or generate MATLAB code that captures the work you’ve done.
You can export an optimization from the GUI as commented code that can be called from the command line and used to automate routines and preserve your work.

MATLAB file of an optimization created using the automatic code generation feature in the Optimization Tool. You can export an optimization from the GUI as commented code that can be called from the command line and used to automate routines and preserve your work.

Global Search and Multistart Solvers

The global search and multistart solvers use gradient-based methods to return local and global minima. Both solvers start a local solver (in Optimization Toolbox) from multiple starting points and store local and global solutions found during the search process.

The global search solver:

- Uses a scatter-search algorithm to generate multiple starting points
- Filters nonpromising start points based upon objective and constraint function values and local minima already found
- Runs a constrained nonlinear optimization solver to search for a local minimum from the remaining start points

The multistart solver uses either uniformly distributed start points within predefined bounds or user-defined start points to find multiple local minima, including a single global minimum if one exists. The multistart solver runs the local solver from all starting points and can be run in serial or in parallel (using Parallel Computing).
Toolbox™). The multistart solver also provides flexibility in choosing different local nonlinear solvers. The available local solvers include unconstrained nonlinear, constrained nonlinear, nonlinear least-squares, and nonlinear least-squares curve fitting.

**Genetic Algorithm Solver**

The genetic algorithm solves optimization problems by mimicking the principles of biological evolution, repeatedly modifying a population of individual points using rules modeled on gene combinations in biological reproduction. Due to its random nature, the genetic algorithm improves your chances of finding a global solution. It enables you to solve unconstrained, bound-constrained, and general optimization problems, and it does not require the functions to be differentiable or continuous.

The following table shows the standard genetic algorithm options provided by Global Optimization Toolbox.

<table>
<thead>
<tr>
<th>Step</th>
<th>Genetic Algorithm Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation</td>
<td>Uniform, feasible</td>
</tr>
<tr>
<td>Fitness scaling</td>
<td>Rank-based, proportional, top (truncation), shift linear</td>
</tr>
<tr>
<td>Selection</td>
<td>Roulette, stochastic uniform selection (SUS), tournament, uniform, remainder</td>
</tr>
<tr>
<td>Crossover</td>
<td>Arithmetic, heuristic, intermediate, scattered, single-point, two-point</td>
</tr>
<tr>
<td>Mutation</td>
<td>Adaptive feasible, Gaussian, uniform</td>
</tr>
<tr>
<td>Plotting</td>
<td>Best fitness, best individual, distance among individuals, diversity of population, expectation of individuals, max constraint, range, selection index, stopping conditions</td>
</tr>
</tbody>
</table>

Global Optimization Toolbox also lets you specify:

- Population size
- Number of elite children
- Crossover fraction
- Migration among subpopulations (using ring topology)
- Bounds, linear, and nonlinear constraints for an optimization problem

You can customize these algorithm options by providing user-defined functions and represent the problem in a variety of data formats, for example by defining variables that are integers, mixed integers, categorical, or complex.

You can base the stopping criteria for the algorithm on time, stalling, fitness limit, or number of generations. And you can vectorize your fitness function to improve execution speed or execute the objective and constraint functions in parallel (using Parallel Computing Toolbox).
Output that shows solutions reached when using only the genetic algorithm (right, bar chart) and when using the genetic algorithm with a gradient-based solver from Optimization Toolbox (final point in Optimization Tool, left). Combining algorithms can produce more accurate results while reducing the number of function evaluations required by the genetic algorithm alone.

**Multiobjective Genetic Algorithm Solver**

Multiobjective optimization is concerned with the minimization of multiple objective functions that are subject to a set of constraints. The multiobjective genetic algorithm solver is used to solve multiobjective optimization problems by identifying the Pareto front—the set of evenly distributed nondominated optimal solutions. You can use this solver to solve smooth or nonsmooth optimization problems with or without bound and linear constraints. The multiobjective genetic algorithm does not require the functions to be differentiable or continuous.

The following table shows the standard multiobjective genetic algorithm options provided by Global Optimization Toolbox.

<table>
<thead>
<tr>
<th>Step</th>
<th>Multiobjective Genetic Algorithm Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation</td>
<td>Uniform, feasible</td>
</tr>
<tr>
<td>Fitness scaling</td>
<td>Rank-based, proportional, top (truncation), linear scaling, shift</td>
</tr>
<tr>
<td>Selection</td>
<td>Tournament</td>
</tr>
<tr>
<td>Crossover</td>
<td>Arithmetic, heuristic, intermediate, scattered, single-point, two-point</td>
</tr>
</tbody>
</table>
### Mutation
Adaptive feasible, Gaussian, uniform

### Plotting
Average Pareto distance, average Pareto spread, distance among individuals, diversity of population, expectation of individuals, Pareto front, rank histogram, selection index, stopping conditions

Global Optimization Toolbox also lets you specify:

- Population size
- Crossover fraction
- Pareto fraction
- Distance measure across individuals
- Migration among subpopulations (using ring topology)
- Linear and bound constraints for an optimization problem

You can customize these algorithm options by providing user-defined functions and represent the problem in a variety of data formats, for example by defining variables that are integers, mixed integers, categorical, or complex.

You can base the stopping criteria for the algorithm on time, fitness limit, or number of generations. And you can vectorize your fitness function to improve execution speed or execute the objective functions in parallel (using Parallel Computing Toolbox).
**Pattern Search Solver**

Global Optimization Toolbox contains three direct search algorithms: generalized pattern search (GPS), generating set search (GSS), and mesh adaptive search (MADS). While more traditional optimization algorithms use exact or approximate information about the gradient or higher derivatives to search for an optimal point, these algorithms use a pattern search method that implements a minimal and maximal positive basis pattern. The pattern search method handles optimization problems with nonlinear, linear, and bound constraints, and does not require functions to be differentiable or continuous.

The following table shows the pattern search algorithm options provided by Global Optimization Toolbox. You can change any of the options from the command line or the Optimization Tool.

<table>
<thead>
<tr>
<th>Pattern Search Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polling methods</td>
<td>Decide how to generate and evaluate the points in a pattern and the maximum number of points generated at each step. You can also control the polling order of the points to improve efficiency.</td>
</tr>
<tr>
<td>Search methods</td>
<td>Choose an optional search step that may be more efficient than a poll step. You can perform a search in a pattern or in the entire search space. Global search methods, like the genetic algorithm, can be used to obtain a good starting point.</td>
</tr>
<tr>
<td>Mesh</td>
<td>Control how the pattern changes over iterations and adjusts the mesh for problems that vary in scale across dimensions. You can choose the initial mesh size, mesh refining factor, or mesh contraction factor. The mesh accelerator speeds up convergence when it is near a minimum.</td>
</tr>
<tr>
<td>Cache</td>
<td>Store points evaluated during optimization of computationally expensive objective functions. You can specify the size and tolerance of the cache that the pattern search algorithm uses and vary the cache tolerance as the algorithm proceeds, improving optimization speed and efficiency.</td>
</tr>
<tr>
<td>Nonlinear constraint algorithm settings</td>
<td>Specify a penalty parameter for the nonlinear constraints as well as a penalty update factor.</td>
</tr>
</tbody>
</table>
Using the Optimization Tool (top) to find the peak, or global optima, of the White Mountains (middle and bottom) using pattern search.

**Simulated Annealing Solver**

Simulated annealing solves optimization problems using a probabilistic search algorithm that mimics the physical process of annealing, in which a material is heated and then the temperature is slowly lowered to decrease defects, thus minimizing the system energy. By analogy, each iteration of a simulated annealing algorithm seeks to improve the current minimum by slowly reducing the extent of the search.

The simulated annealing algorithm accepts all new points that lower the objective, but also, with a certain probability, points that raise the objective. By accepting points that raise the objective, the algorithm avoids being trapped in local minima in early iterations and is able to explore globally for better solutions.

Simulated annealing allows you to solve unconstrained or bound-constrained optimization problems and does not require that the functions be differentiable or continuous. From the command line or Optimization Tool you can use toolbox functions to:

- Solve problems using adaptive simulated annealing, Boltzmann annealing, or fast annealing algorithms
- Create custom functions to define the annealing process, acceptance criteria, temperature schedule, plotting functions, simulation output, or custom data types
- Perform hybrid optimization by specifying another optimization method to run at defined intervals or at normal solver termination
Using simulated annealing to solve a challenging problem that contains flat regions between basins.

**Solving Optimization Problems Using Parallel Computing**

You can use Global Optimization Toolbox in conjunction with Parallel Computing Toolbox to solve problems that benefit from parallel computation. By using built-in parallel computing capabilities or defining a custom parallel computing implementation of your optimization problem, you decrease time to solution.

Built-in support for parallel computing accelerates the objective and constraint function evaluation in genetic algorithm, multiobjective genetic algorithm, and pattern search solvers. You can accelerate the multistart solver by distributing the multiple local solver calls across multiple MATLAB workers or by enabling the parallel gradient estimation in the local solvers.
Demonstration of Using the Genetic Algorithm with Parallel Computing Toolbox

A custom parallel computing implementation involves explicitly defining the optimization problem to use parallel computing functionality. You can define either your objective function or constraint function to use parallel computing, letting you decrease the time required to evaluate the objective or constraint.