Metaheuristics: A Julia Package for Single- and Multi-Objective Optimization

Jesús-Adolfo Mejía-de-Dios and Efrén Mezura-Montes

1 Artificial Intelligence Research Institute, University of Veracruz, MEXICO

Summary

Metaheuristics is a Julia package that implements metaheuristic algorithms for solving global optimization problems that can contain constraints and either single or multiple objectives. The package exposes an easy-to-use API to enable testing without requiring lengthy configuration for choosing among the implemented optimizers. For example, to optimize an objective function $f(x)$ where solution $x$ is within the corresponding bounds using the Evolutionary Centers Algorithm optimizer, the user can call `optimize(f, bounds, ECA())`. Moreover, Metaheuristics provides common features required by the evolutionary computing community such as challenging test problems, performance indicators, and other notable utility functions. This paper presents the main features followed by some examples to illustrate the usage of this package for optimization.

Statement of need

Real-world problems often require sophisticated methods to solve. Metaheuristics are stochastic algorithms that approximate optimal solutions quickly where not all of the mathematical properties of the problem are known. Our package implements state-of-the-art algorithms for constrained, multi-, and many-objective optimization. It also includes many other utility functions such as performance indicators, scalable benchmark problems, constraint handling techniques, and multi-criteria decision-making methodologies. Although similar software has been proposed in different programming languages such as Python (Blank & Deb, 2020), MATLAB (Tian et al., 2017), C/C++ (Biscani & Izzo, 2020), and Java (Nebro et al., 2015), among others (Johnson, 2022). Metaheuristics is the first package in Julia containing ready-to-use metaheuristic algorithms and utility functions with a uniform API.

There are also some packages implemented in Julia for global optimization. For example, OptiM.jl (Mogensen & Riseth, 2018) implements global optimizers such as Particle Swarm Optimization, BlackBoxOptim.jl (Feldt, 2022) implements a couple of stochastic heuristics for black-box optimization, Evolutionary.jl (Wilde et al., 2021) is a framework for evolutionary computing, and CMAEvolutionStrategy.jl (Brea, 2022) implements a CMA Evolution Strategy. Unlike these packages, Metaheuristics can handle equality and inequality constraints and supports multi-objective problems.

Main Features

This section describes the primary features of Metaheuristics. First, Metaheuristics implements a consistent and intuitive API to approximate the optimal solutions of the following minimization problem:
\[
\min_{x \in X} f(x)
\]
subject to
\[
g_j(x) \leq 0, \ j = 1, 2, \ldots, J;
\]
\[
h_l(x) = 0, \ l = 1, 2, \ldots, L;
\]
where \(x_{i,\text{min}} \leq x_i \leq x_{i,\text{max}}\), i.e., \(x \in X = \prod_{i=1}^{D} [x_{i,\text{min}}, x_{i,\text{max}}]\). \(f\) can be either single- or multi-objective.

### Implemented Metaheuristics

Implemented metaheuristic algorithms are detailed in Table 1. Algorithms for single-objective optimization include Evolutionary Centers Algorithm (ECA) (Mejía-de-Dios & Mezura-Montes, 2019), Differential Evolution (DE) (Price, 2013), Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Artificial Bee Colony (ABC) (Karaboga & Basturk, 2007), Gravitational Search Algorithm (GSA) (Mirjalili & Gandomi, 2017), Simulated Annealing (SA) (Van L. & Aarts, 1987), Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016), and Machine Coded Compact Genetic Algorithms (MCCGA) (Satman & Akadal, 2020). Metaheuristics also includes multi-objective optimization algorithms such as a Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D-DE) (Li & Zhang, 2008), Non-dominated Sorting Genetic Algorithms (NSGA-II,-III) (Deb et al., 2002; Deb & Jain, 2014), S-Metric Selection Evolutionary Multi-objective Algorithm (SMS-EMOA) (Emmerich et al., 2005), Improved Strength Pareto Evolutionary Algorithm (SPEA2) (Eckart Zitzler et al., 2001) and Coevolutionary Framework for Constrained Multiobjective Optimization (CCMO) (Tian et al., 2021).

### Table 1: Implemented optimizers so far. Here, “✓”, “×” and “−”, respectively mean that the feature in the corresponding column is available, unavailable, or can be enabled by changing default parameters.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Objective</th>
<th>Constraint Handling</th>
<th>Batch Evaluation</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECA</td>
<td>Single</td>
<td>✓</td>
<td>✓</td>
<td>Mejía-de-Dios &amp; Mezura-Montes (2019)</td>
</tr>
<tr>
<td>DE</td>
<td>Single</td>
<td>✓</td>
<td>✓</td>
<td>Price (2013)</td>
</tr>
<tr>
<td>ABC</td>
<td>Single</td>
<td>×</td>
<td>×</td>
<td>Karaboga &amp; Basturk (2007)</td>
</tr>
<tr>
<td>GSA</td>
<td>Single</td>
<td>×</td>
<td>✓</td>
<td>Mirjalili &amp; Gandomi (2017)</td>
</tr>
<tr>
<td>WOA</td>
<td>Single</td>
<td>✓</td>
<td>✓</td>
<td>Mirjalili &amp; Gandomi (2017)</td>
</tr>
<tr>
<td>MCCGA</td>
<td>Single</td>
<td>×</td>
<td>×</td>
<td>Satman &amp; Akadal (2020)</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>Multi</td>
<td>✓</td>
<td>✓</td>
<td>Deb et al. (2002)</td>
</tr>
<tr>
<td>SMS-EMOA</td>
<td>Multi</td>
<td>✓</td>
<td>✓</td>
<td>Emmerich et al. (2005)</td>
</tr>
<tr>
<td>SPEA2</td>
<td>Multi</td>
<td>✓</td>
<td>✓</td>
<td>Eckart Zitzler et al. (2001)</td>
</tr>
<tr>
<td>CCMO</td>
<td>Multi</td>
<td>✓</td>
<td>✓</td>
<td>Tian et al. (2021)</td>
</tr>
<tr>
<td>NSGA-III</td>
<td>Many</td>
<td>✓</td>
<td>✓</td>
<td>Deb &amp; Jain (2014)</td>
</tr>
</tbody>
</table>

### Performance Indicators

Performance indicators are used to assess algorithms with respect to the quality of the corresponding outcomes (E. Zitzler et al., 2003). Metaheuristics implements Generational Distance (GD), Inverted Generational Distance (IGD), IGD+., Covering Indicator (C-metric),

Hypervolume (HV), Averaged Hausdorff distance (also known as \(\Delta_p\)), Spacing Indicator, and the \(\varepsilon\)-indicator.

**Multi-Criteria Decision-Making**

Metaheuristics also provides algorithms for when Multi-Criteria Decision-Making (MCDM) has to be performed after an optimizer has reported a set of non-dominated solutions. The implemented MCDM techniques include Compromise Programming (Ringuest, 1992) and Region of Interest Archiving (de-la-Cruz-Martínez et al., 2023). Metaheuristics also includes an interface to the JMcDM package (Satman et al., 2021), which implements many further MCDM techniques.

**Installation and Usage**

Metaheuristics can be installed through the Julia package manager by using the add command in the Pkg prompt:

```
pkg> add Metaheuristics
```

Or, equivalently, via the Pkg API:

```
julia> import Pkg
julia> Pkg.add("Metaheuristics")
```

Now, let us consider the following minimization problem:

Minimize:

\[
f(x) = 10D + \sum_{i=1}^{D} x_i^2 - 10 \cos(2\pi x_i)
\]

with box constraints (bounds) \(x \in [-10, 10]^5\).

We can optimize the objective function \(f\) with solution \(x\) within the desired bounds as follows:

```julia
julia> f(x) = 10length(x) + sum( x .^2 - 10cos.(2π*x) )

julia> bounds = [-10ones(5) 10ones(5)]

julia> result = optimize(f, bounds, DE(CR=0.5))
```

+=========== RESULT ==========+
iteration: 550
minimum: 0
minimizer: [-1.5469066028117595e-9,..., 3.797322900567224e-9]
f calls: 27500
total time: 0.0388 s
stop reason: Small difference of objective function values.
+============================+

optimize(f, bounds, OPTIMIZER) is used to approximate an optimal solution, where OPTIMIZER can be selected from the implemented metaheuristics (see Table 1). Note that OPTIMIZER is a stochastic procedure and therefore each run may show different outputs.

Finally, potential users are encouraged to read the documentation for more details, options, and examples.

**Acknowledgements**

The first author acknowledges support from the Mexican Council for Science and Technology (CONACyT) through a scholarship to pursue graduate studies at the University of Veracruz.
MEXICO.

References


