

MOEA Framework Quick Start

A Free and Open Source Java Framework for Multiobjective Optimization

David Hadka
Version 2.12

1 What is the MOEA Framework?

The MOEA Framework is a free and open source Java library for developing and experimenting with multiobjective evolutionary algorithms (MOEAs) and other general-purpose optimization algorithms. A number of algorithms are provided out-of-the-box, including NSGA-II, NSGA-III, ϵ -MOEA, GDE3 and MOEA/D. In addition, the MOEA Framework provides the tools necessary to rapidly design, develop, execute and statistically test optimization algorithms. The following features of the MOEA Framework distinguish it from available alternatives:

Fast, reliable implementations of many state-of-the-art multiobjective evolutionary algorithms. The MOEA Framework contains internally NSGA-II, NSGA-III, ϵ -MOEA, ϵ -NSGA-II, PAES, PESA2, SPEA2, IBEA, SMS-EMOA, GDE3, SMPSO, OMOPSO, CMA-ES, and MOEA/D. These algorithms are optimized for performance, making them readily available for high performance applications. By also supporting the JMetal and PISA libraries, the MOEA Framework provides access to 30 multiobjective optimization algorithms.

Extensible with custom algorithms, problems and operators. The MOEA Framework provides a base set of algorithms, test problems and search operators, but can also be easily extended to include additional components. Using a Service Provider Interface (SPI), new algorithms and problems are seamlessly integrated within the MOEA Framework.

Modular design for constructing new optimization algorithms from existing components. The well-structured, object-oriented design of the MOEA Framework library allows combining existing components to construct new optimization algorithms. And if needed functionality is not available in the MOEA Framework, you can always extend an existing class or add new classes to support any desired feature.

Permissive open source license. The MOEA Framework is licensed under the free and open GNU Lesser General Public License, version 3 or (at your option) any later version. This allows end users to study, modify, and distribute the MOEA Framework freely.

Fully documented source code. The source code is fully documented and is frequently updated to remain consistent with any changes. Furthermore, an extensive user manual is provided detailing the use of the MOEA Framework in detail.

Extensive support available online. As an actively maintained project, bug fixes and new features are constantly added. We are constantly striving to improve this product. To aid this process, our website provides the tools to report bugs, request new features, or get answers to your questions.

Over 1200 test cases to ensure validity. Every release of the MOEA Framework undergoes extensive testing and quality control checks. And, if any bugs are discovered that survive this testing, we will promptly fix the issues and release patches.

2 Which Distribution is Right for Me?

The MOEA Framework is currently distributed in three forms: 1) the compiled binaries; 2) the source code; and 3) the demo application. The following text describes each distribution and its intended audience.

Compiled Binaries The compiled binaries distribution contains a fully-working MOEA Framework installation. All required third-party libraries, data files and documentation are provided. This download is recommended for developers integrating the MOEA Framework into an existing project.

Source Code The source code distribution contains all source code, unit tests, documentation and data files. This distribution gives users full control over the MOEA Framework, as any component can be modified as needed. As such, this download is recommended for developers wishing to contribute to or study the inner workings of the MOEA Framework.

Demo Application The demo application provides several interactive demos of the MOEA Framework launched by double-clicking the downloaded JAR file. This download is intended for first-time users to quickly learn about the MOEA Framework and its capabilities.

3 Obtaining a Copy

The various MOEA Framework distributions can be downloaded from our website at moeaframework.org/. The compiled binaries and source code distributions are packaged in a compressed tar (.tar.gz) file. Unix/Linux/Mac users can extract the file contents using the following command:

```
tar -xzf MOEAFramework-2.12.tar.gz
```

Windows users must use an unzip utility like 7-Zip to extract the file contents. 7-Zip is a free, open source program which can be downloaded from <http://www.7-zip.org/>.

4 Installing Dependencies

The software packages listed below are required or recommended in order to use the MOEA Framework. Any software package marked as required **MUST** be installed on your computer in order to use the MOEA Framework. Software marked as optional is not required to be installed, but will generally make your life easier.

4.1 Java 6+ (Required)

Java 6, or any later version, is required for any system running the MOEA Framework. If downloading the compiled binaries or demo application, you only need to install the Java Runtime Environment (JRE). The source code download requires the Java Development Kit (JDK), which

contains the compiler and other developer tools. We recommend one of the following vendors (most are free):

Oracle - <http://www.oracle.com/technetwork/java/javase/>

- For Windows, Linux and Solaris

JRockit JDK - <http://www.oracle.com/technetwork/middleware/jrockit/>

- For Windows, Linux and Solaris
- May provide better performance and scalability on Intel 32 and 64-bit architectures

OpenJDK - <http://openjdk.java.net/>

- For Ubuntu 8.04 (or later), Fedora 9 (or later), Red Hat Enterprise Linux 5, openSUSE 11.1, Debian GNU/Linux 5.0 and OpenSolaris

IBM - <http://www.ibm.com/developerworks/java/jdk/>

- For AIX, Linux and z/OS

Apple - <http://support.apple.com/kb/DL1572>

Please follow the installation instruction accompanying your chosen JRE or JDK.

4.2 Eclipse or NetBeans (Optional)

Eclipse and NetBeans are two development environments for writing, debugging, testing, and running Java programs. Eclipse can be downloaded for free from eclipse.org/, and NetBeans can be obtained from netbeans.org/.

The installation of Eclipse is simple — just extract the compressed file to a folder of your choice and run the Eclipse executable from this folder. First-time users of Eclipse may be prompted to select a workspace location. The default location is typically fine. Click the checkbox to no longer show this dialog and click Ok.

To install NetBeans, simply run the downloaded executable. Once installed, you can launch NetBeans by clicking the NetBeans link in your start menu.

4.3 Apache Ant (Optional)

Apache Ant is a Java tool for automatically compiling and packaging projects, similar to the Make utility on Unix/Linux. Individuals working with the source code distribution should consider installing Apache Ant, as it helps automate building and testing the MOEA Framework. Apache Ant can be downloaded from <http://ant.apache.org/>. The installation instructions provided by Ant should be followed.

Note that Eclipse contains Ant, so it is not necessary to install Eclipse and Ant together.

5 Importing into Eclipse

When working with the source code distribution, it is necessary to properly configure the Java environment to ensure all resources are available. To assist in this process, the source code distribution includes the necessary files to import directly into Eclipse.

To import the MOEA Framework project into Eclipse, first start Eclipse and select File → Import... from the menu. A popup window will appear. Ensure the General → Existing Projects into Workspace item is selected and click Next. A new window will appear. In this new window, locate the Set Root Directory entry. Using the Browse button, select the MOEAFramework-2.12 folder. Finally, click Finish. The MOEA Framework will now be properly configured in Eclipse.

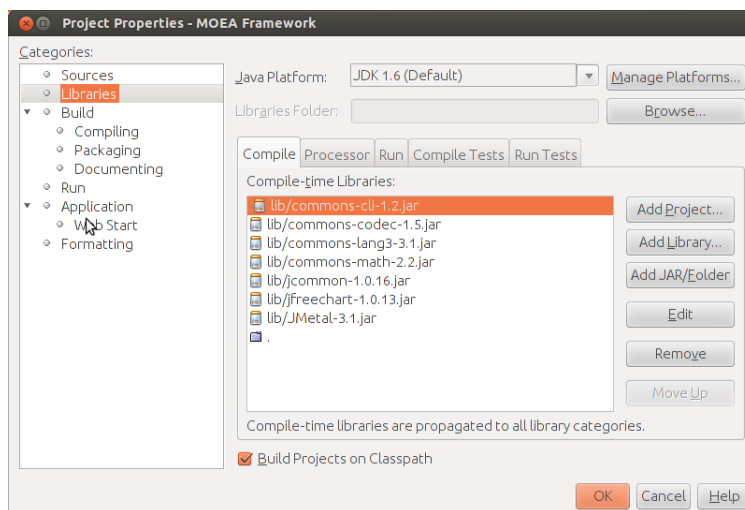
6 Importing into NetBeans


If you downloaded the source code, you can import the MOEA Framework into NetBeans as follows. In NetBeans, select New Project from the File menu. In the screen that appears, select the “Java” category and “Java Project with Existing Sources”. Click Next.

Specify the project name as “MOEA Framework”. Set the project folder by clicking the Browse button and selecting the MOEAFramework-2.12 folder. Click Next.

Add the src and examples folders as Source Package Folders. Click Finish. The MOEA Framework project should now appear in the Projects window.

Finally, we need to add the third-party libraries used by the MOEA Framework. Right-click the MOEA Framework project in the Projects window and select Properties. In the window that appears, click Libraries in the left-hand panel. On the right-side of the window, click the button “Add Jars/Folder”. Browse to the MOEAFramework-2.12/lib folder, highlight all the JAR files (using shift or alt to select multiple files), and click Ok. Be sure that you select each individual JAR file and not the folder containing the JAR files. Click the “Add Jars/Folder” button again. Navigate to and select the root MOEAFramework-2.12 folder, and click Ok. You should now see 8 items in the compile-time libraries list. There should be 7 entries referencing .jar files the “ .” as the last entry. Your screen should look like the figure below. Click Ok when finished.



Test your NetBeans install by running Example1. You can run an example by expanding the  examples folder in the Project window, right-clicking Example1, and selecting Run File from the popup menu.

7 Resolving Dependencies with Maven

Since version 2.4, the MOEA Framework and its dependencies can be resolved using the Maven dependency management system. Maven is available from <http://maven.apache.org/>. To add the MOEA Framework to your Maven project, add the following dependency to your `pom.xml` file:

```
<dependency>
  <groupId>org.moeaframework</groupId>
  <artifactId>moeaframework</artifactId>
  <version>2.12</version>
</dependency>
```

If this is your first time using Maven, check out the Maven in 5 Minutes guide at <http://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>.

8 Testing your Installation

Having finished installing the MOEA Framework and its dependencies, it is useful to run the MOEA Diagnostic Tool to test if the installation was successful. If the diagnostic tool appears and you can run any algorithm, then the installation was successful.

Compiled Binaries Run the `launch-diagnostic-tool.bat` file on Windows. You can manually run the diagnostic tool with the following command:

```
java -Djava.ext.dirs=lib
      org.moeaframework.analysis.diagnostics.LaunchDiagnosticTool
```

Source Code Inside Eclipse, navigate to the `src` → `org` → `moeaframework` → `analysis` → `diagnostic` package in the Package Explorer window. Right-click the file `LaunchDiagnosticTool.java` and select the Run as → Java Application option in the popup menu.

Demo Program Double-click the downloaded JAR file. If the demo window does not appear, try to manually launch the tool with the following command:

```
java -jar MOEAFramework-2.12-Demo.jar
```

9 Running Examples

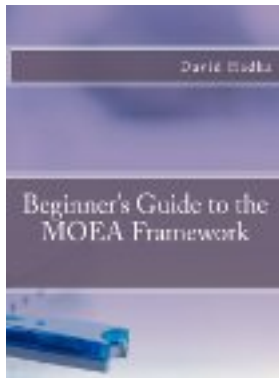
Several example applications are provided with the MOEA Framework in the `MOEAFramework-2.12/examples/` folder. If using Eclipse or Netbeans, you can run these examples directly within the GUI. For example, in Eclipse, right-click `Example1.java` and select `Run as → Java Application`. If running from the command line, you will first need to compile the code using the Java compiler (`javac`) followed by running the example in the Java virtual machine (`java`). For example, you can compile and run the first example on Windows as follows:

```
javac -cp ".;examples;lib/*" examples/Example1.java
java -cp ".;examples;lib/*" Example1
```

If using Unix/Linux, replace the semicolons (;) with colons (:) in the above commands. You can find more details on the available examples on our website at moeaframework.org/examples.html or viewing the source code for the examples.

10 Getting Help

As a new user to the MOEA Framework, you may find it helpful to walk through our examples. As demonstrated in the previous section, you can find and run our examples in the `MOEAFramework-2.12/examples/` folder.



Additionally, we have authored two helpful books for purchase from moeaframework.org/documentation.html. The *Beginner's Guide to the MOEA Framework* covers topics ranging from solving unconstrained and constrained optimization problems, comparing the performance of algorithms, plotting results, developing custom algorithms, integrating your own optimization problems, and more. A supplemental book covers developing distributed and high-performance parallel algorithms built using master-slave, island, and hybrid architectures. Both books include complete, downloadable code examples to get you started quickly.

Advanced users who wish to learn the inner workings of the MOEA Framework or make significant modifications can find the API specification available on our website at moeaframework.org/documentation.html. This specification details every class and method provided by the MOEA Framework, and often includes examples or detailed descriptions of their usage.

Lastly, it is common for users to have questions or requests not answered by these resources. Please visit our Github page at <https://github.com/MOEAFramework/MOEAFramework>. We recommend searching the issue tracker to see if a user previously asked your question. You can also create issues for any bugs or odd behavior you have encountered.

11 Conclusion

Thanks again for downloading the MOEA Framework. We hope you find it useful in your work. If using the MOEA Framework in academic publications, please include a citation as follows:

Hadka, David (2015). MOEA Framework - A Free and Open Source Java Framework for Multiobjective Optimization. Version 2.12, <http://www.moeaframework.org/>.