OAT: A Quick Start Guide

JASON BROWNLEE
Guide 20071220A
Complex Intelligent Systems Laboratory, Centre for Information Technology Research,
Faculty of Information and Communication Technologies, Swinburne University of Technology
Melbourne, Australia
jbrownlee AT ict.swin.edu.au
or
jbrownlee AT users.sourceforge.net

I. INTRODUCTION

The *Optimization Algorithm Toolkit* (OAT) is an open source software project written in Java that provides a suite of Computational Intelligence optimization problem domains with problem instances, classical and state-of-the-art algorithms, visualisation, graphs, and much more. This work provides a quick start guide for *using the OAT* (Section II.) and *developing for the OAT* (Section III.). This work considers OAT 1.4 as of December 2007 that may be accessed via the OAT Software webpage `http://optalgtoolkit.sourceforge.net` and the OAT Project webpage `http://sourceforge.net/projects/optalgtoolkit`. For information regarding the OAT software overview, vision, and future see Brownlee [2], and for information regarding development of new problem domains, problem instances and or algorithm instances, see Brownlee [1].

II. OAT USAGE

This section provides rudimentary instructions for getting started with *using* the OAT.

A. *Online*

The *OAT Explorer* interface may be used directly through a web browser. Applet versions of most of the base problem domains and all related problem and algorithm instances are provided on the OAT Software webpage.

1. *Locate, Download, and Install the latest version of the Sun Java Runtime Environment (JRE).*

OAT requires Java to execute. If you have already installed the Sun JDK then this step is not needed as the JRE is installed when you installed the Sun JDK. The latest version of the Sun JRE is available from the Java webpage (`http://www.java.com`) and or the Sun java webpage (`http://www.java.sun.com`).

2. *Access the OAT Software webpage and select a specific problem domain.*

The OAT applets may be accessed from the OAT Software webpage (`http://optalgtoolkit.sourceforge.net`) under the '*use online*' section. Applets are provided for each available problem domain, as well as a master applet that combines all available domains into a single application. A given applet may take some time to load as the program and its dependant libraries must be downloaded before being executed by the Java runtime within the browser. If this load time is too long, try using OAT in the offline mode.

B. *Offline*

The OAT was designed to be used offline that means that the software must be downloaded and installed before it is executed. The offline experience is richer than the online experience as it provides more problem domains, faster load times, and an experimenter interface.

1. *Locate, Download, and Install the latest version of the Sun Java Runtime Environment (JRE).*

OAT requires Java to execute. If you have already installed the Sun JDK then this step is not needed as the JRE is installed when you installed the Sun JDK. The latest version of the Sun JRE is available from the Java webpage (`http://www.java.com`) and or the Sun java webpage (`http://www.java.sun.com`).

2. *Locate, Download, and Install the OAT.*

OAT can be downloaded from the OAT Project website's downloads page (`http://downloads.sourceforge.net/optalgtoolkit`). OAT is distributed as a zip archive. Given that OAT is platform independent, the same distribution is used regardless of the operating system you use, for example Microsoft Windows, Apple OSX, and flavours of Unix and Linux. To install OAT use a decompression program to unzip the OAT distribution, common examples include *WinZip* on Microsoft Windows (`http://www.winzip.com`), *Stuffit* on Apple OSX (`http://www.stuffit.com`), and GNU *gzip* on Linux distributions. Unzip OAT into the common location on your hard disk drive where your other applications are installed, for example `c:/Program Files/OAT` on Microsoft Windows, `Applications\OAT` on Apple OSX, and `~\OAT` on Linux distributions.

3. *Execute OAT.*

There are a number of ways to execute OAT and a number of user entry points. The *OAT Launcher* provides a generic entry point into the OAT graphical user interfaces (GUI's) allowing a user to select and interchange between the *OAT Explorer* and *OAT Experimenter*. To execute the OAT Launcher in a graphical environment such as Microsoft Windows, Apple OSX, and X11, one may double-click the executable jar file (`optalgtoolkit.jar`).

To execute the OAT Launcher a command line environment issue one of the following commands:

```
>java –jar optalgtoolkit.jar
```

or

```
>java –classpath optalgtoolkit.jar com.oat.gui.launcher.LauncherMain
```

The OAT Explorer may be launched directed. For example, one may execute an instance of the OAT Explorer with all available problem domains as follows:

```
>java –classpath optalgtoolkit.jar com.oat.explorer.gui.entry.ExplorerMain
```

One may execute an instance of the OAT Explorer for each specific problem domain, for example the explorer for the Continuous Function Optimization (CFO) domain:

```
>java –classpath optalgtoolkit.jar com.oat.explorer.domains.cfo.gui.entry.
CFOMainFrame
```

The OAT Experimenter may be launcher directly. For example, one may execute an instance of the OAT Experimenter with all available problem domains as follows:

```
>java –classpath optalgtoolkit.jar
com.oat.experimenter.gui.entry.ExperimenterMain
```

OAT may require large amounts of memory for some problems or applications. Java may be tuned to allocate more memory and other performance improvements (for example see

http://java.sun.com/performance/reference/whitepapers/tuning.html). An example of tuning Java to exploit 256 Megabytes of Random Access Memory (RAM) and two Central Processing Units (CPU's) is provided as follows:

```
java -XX:ParallelGCThreads=2 -XX:+UseParallelGC –Xms64M -Xmx250M -server -
jar optalgtoolkit.jar
```

III. OAT DEVELOPMENT

This section provides rudimentary instructions for getting started with the development with OAT. This section assumes the Eclipse Integrated Development Environment (IDE) will be used for development.

1. *Locate, Download, and Install the latest version of the Sun Java Runtime Environment (JRE) or the Java Development Kit (JDK).*

OAT requires Java to compile and to execute. The latest version of the Sun JRE is available from the Java webpage (http://www.java.com), and or the latest version of the Sun JDK and Sun JRE are available from the Sun java webpage (http://www.java.sun.com).

2. *Locate, Download, and Install the latest version of the Eclipse Integrated Development Environment (IDE).*

The Eclipse IDE was used in the development of the OAT and is recommended for software development on or with the platform. The latest version of the *Eclipse IDE for Java Developers* may be downloaded from the Eclipse website (http://www.eclipse.org).

3. *Acquire the OAT source code and dependency libraries.*

The OAT source code and dependency libraries may be acquired either through the Concurrent Version System (CVS), which is the preferred method, or by downloading the OAT distribution from the OAT Project webpage (the source is located in the distribution within an archive with the name optalgtoolkitX.X-src.zip where 'X_X' refers to the version number of the source code). The Eclipse IDE has support for the creating a new project based on a source tree stored in a CVS repository. Refer to the Eclipse IDE documentation for the appropriate procedure for this approach (http://help.eclipse.org). Refer to the OAT Project webpage for information for anonymous developer CVS access to the OAT source tree (http://sourceforge.net/cvs/?group_id=182624). Each release of OAT is tagged with the version number using the convention: optalgtoolkitX_X, where 'X_X' referees to the version number.

4. *Configure and test the project.*

The OAT is dependant on third party libraries that must be included in the project build path. All required dependency libraries are provided in a /lib subdirectory with OAT both in the CVS and distribution release. Libraries may be added to the created Java project in Eclipse via the project properties (http://help.eclipse.org). The present version of OAT is depended upon the following third party libraries: JFreeChart (http://www.jfree.org), Open Source Physics (http://www.opensourcephysics.org), JUnit (http://junit.org), The Huygens Webservices Client (http://gungurru.csse.uwa.edu.au/cgi-bin/WebObjects/huygensWS.woa), Apache Jakarta Commons Math (http://commons.apache.org/math/), Colt (http://dsd.lbl.gov/~hoschek/colt), SSJ (http://www.iro.umontreal.ca/~simardr/ssj), and SOCR (http://www.socr.ucla.edu). The configured OAT project may be tested by loading the OAT Launcher from within Eclipse (com.oat.gui.launcher.LauncherMain).

5. *Build the project.*

An *ant* build script is provided to build the OAT project including the distributed version, the webpage, and the Application Programming Interface (API) documentation called Javadoc. The ant build script is located in the base directory within the source tree and is has the name `build.xml`. Any scripts are executed using Apache Ant (`http://ant.apache.org`) and can be executed within Eclipse (refer to Eclipse documentation `http://help.eclipse.org`).

Acknowledgements

Thankyou to Tim Hendtlass for his support. Thankyou for Irene Moser for promoting me to create this document.

Bibliography

[1] Jason Brownlee, "OAT HowTo: High-level Domain, Problem, and Algorithm Implementation," Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, Victoria, Australia, Technical Report 20071218A, Dec 2007.

[2] Jason Brownlee, "OAT: The Optimization Algorithm Toolkit," Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, Victoria, Australia, Technical Report 20071220A, Dec 2007.