General notes:
The LPSolve55 native libraries were downloaded from http://lpsolve.sourceforge.net/5.5/. The versions here have been modified to update the shell files that build the libraries, and the instructions below are based on these updated packages.

The shell files in the next steps HATE paths with spaces in them. If possible, locate the extracted files in a directory with no spaces in its path. Set the JAVA_HOME environment variable to point to the desired Java JDK installation. For the 64-bit libraries I used JDK 17 to align with NetLogo 7.0. For the Linux 32-bit library I used the latest 32-bit JRE I could find, OpenJRE 1.9.

For the Macintosh and Linux libraries:

1. Extract lp_solve5.5.2.11_source. We will want to use the lpsolve55 sub-directory to build liblpsolve55.dylib/.so, depending on the platform.
1. Extract lp_solve5.5.2.11_java. We will use this to build liblpsolve55j.jnilib/.so/, depending on the platform.
2. Go to lp_solve5.5.2.11_source/lpsolve55 and run the appropriate ccc file, ccc.osx, ccc.x86, ccc.lnx64, ccc.lnx32 as appropriate to the desired platform, each as modified by us. This will create liblpsolve55.a and liblpsolve55.dylib/.so in the /bin sub-directory. Since the liblpsolve55 library is needed to build the liblpsolve55j library, this step must be completed before the next.
3. Now go to lp_solve5.5.2.11_java/lib. Open build.osx. build.x86, build.lnx64, build.lnx32, as appropriate, each in the appropriate subdirectory. Check the paths where indicated and modify as needed. Note the -L$LPSOLVE_DIR/lpsolve55/bin/--- option on the g++ command to build the dynamic library. This points to the files that were created in step 2 for the appropriate platform.
4. Copy the header files jni.h and jni_md.h from the Java installation to lp_solve5.5.2.11_java/src/c. I found these files in $JAVA_HOME/include.
5. Run the appropriate build file. This will create liblpsolve55j.jnilib/.so in the same directory.
6. NOTE, that executing these shell files, the compilers will throw lots of warnings about old C++ syntax. For now, they seem to make no difference in the final results and may be ignored.
7. Congratulations!

For the Windows libraries;

1. So far, no success in building them. A work in progress. However, prebuilt versions were available and so have been used in this extension.