

Appendix C

User Guide

This user guide provides a overview of FuxCP, covering its installation process, usage within OpenMusic, and a description of the costs displayed in the interface. While FuxCP is designed to be compatible with all platforms, it relies on GiL, which currently works only on MacOS and Linux. Unfortunately, GiL does not support Windows due to compatibility issues between the 32-bit Lisp license used by OpenMusic and the 64-bit Gecode Windows version. Although it is technically possible to obtain a 32-bit version of Gecode for Windows, it is not recommended.

C.1 Installing FuxCP

C.1.1 Prerequisites

To use FuxCP, it is necessary to download and install the following tools:

- Gecode on <https://www.gecode.org/download.html>
- OpenMusic on <https://openmusic-project.github.io/openmusic/>

And download the following libraries:

- GiL on <https://github.com/sprockeelsd/GiLv2.0>
- FuxCP: <https://github.com/sprockeelsd/Melodizer>

On the last github, other tools such as Melodizer and Melodizer2.0 are available. In the context of this user guide, only the FuxCP folder will be necessary.

C.1.2 Loading FuxCP in OpenMusic

To use the previous libraries, OpenMusic must be launched. Upon opening any workspace, locate the toolbar at the top of the interface. Click on the "Windows" button, highlighted in figure C.1, and select "Library" from the dropdown menu. This action will unveil a new window. In the toolbar of this window, choose "File" and then "Add remote library." Navigate through your file system to find the path where the previously downloaded FuxCP and Gil libraries are stored. Once located, the libraries should appear under the "libraries" folder in the "Library" window, as depicted in figure C.2. Right-click on "fuxcp" and select "Load Library". If no errors occur, the setup is complete.

However, if an error arises, it may be a linking issue with the Gecode library. For MacOS users, a script can be used from the c++ folder of the gil library. Edit the path to Gecode inside the script to match your system's configuration. Linux users should add the Gecode library to the LD_LIBRARY_PATH variable. Go to the `/etc/ld.so.conf.d` folder and create a new `.conf` file if one does not already exist. In this file, paste the full path to the Gecode library, save it, and run `sudo ldconfig` to update the system with the new library. Don't forget to restart OpenMusic and don't stop believing. Following these steps should ensure the proper utilization of FuxCP.

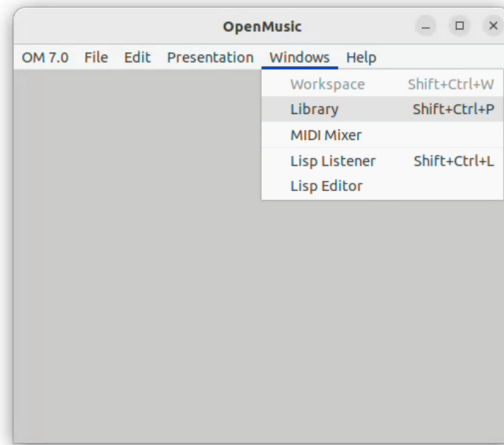


Figure C.1: Opening the "Library" window in OpenMusic.

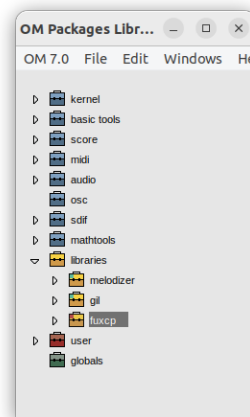


Figure C.2: Loading the "fuxcp" library in OpenMusic.

C.2 Using FuxCP in OpenMusic

It is straightforward to use FuxCP in OpenMusic. There is a single block comprising the entire graphical interface of the tool. This block or class is called `cp-params`. To load it, it is possible to type `fuxcp : cp-params` in a new patch entry; or load the block of the class by loading "cp-params" from the drop-down menu by right-clicking in the patch (*Classes* → *Libraries* → *FuxCP* → *Solver* → *CP – PARAMS*).

Once this block has appeared, all you have to do is bind an OM voice object, representing the *cantus firmus*, to the second argument of `cp-params` as shown in figure C.3. Don't forget to block the input voice object and evaluate `cp-params` so it can detect the new input. Now `cp-params` can be blocked too. From now on, you could directly use the interface and generate counterpoints using the tool. If you want to retrieve the voice object containing the counterpoint generated by the tool, just bind the third argument on the output side to a voice object. Once bound, it is then possible to evaluate the voice object so that it updates.

But how to use the interface? Just double-click on the block to make it appear. The interface is sorted from left to right, so that the preferences are separated into three different categories: "Preferences for Melodic Intervals of...", "General Preferences",

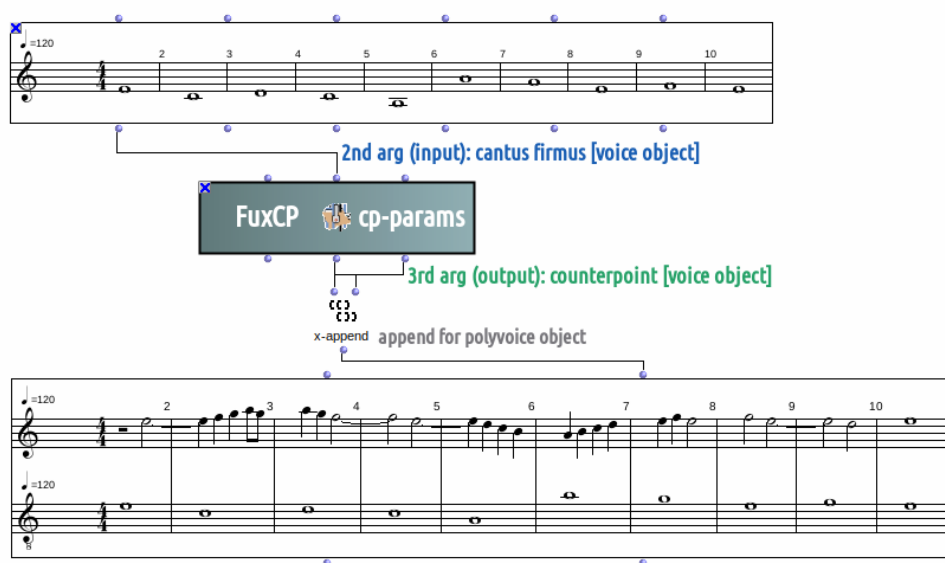


Figure C.3: View of a patch using `fuxcp::cp-params` in OpenMusic.

"Species Specific Preferences", "Solver Configuration", and in the lower right corner, "Solver Launcher" (see figure C.4). Once the preferences have been chosen, the default ones representing the style of Fux, you must save the parameters ("Save Config") in order to then be able to launch the search for a solution ("Next Solution"). This search can take a fraction of a second just as it can take tens of minutes, or even hours if the parameters chosen make the search difficult. If a search takes too long, it is always possible to stop it by clicking on "Stop". You can then either change the preferences in a way (often at the level of the costs of the melodic intervals), or increase the "Irreverence" to obtain potentially less "good" but faster solutions. The description of the parameters is available in the next section.

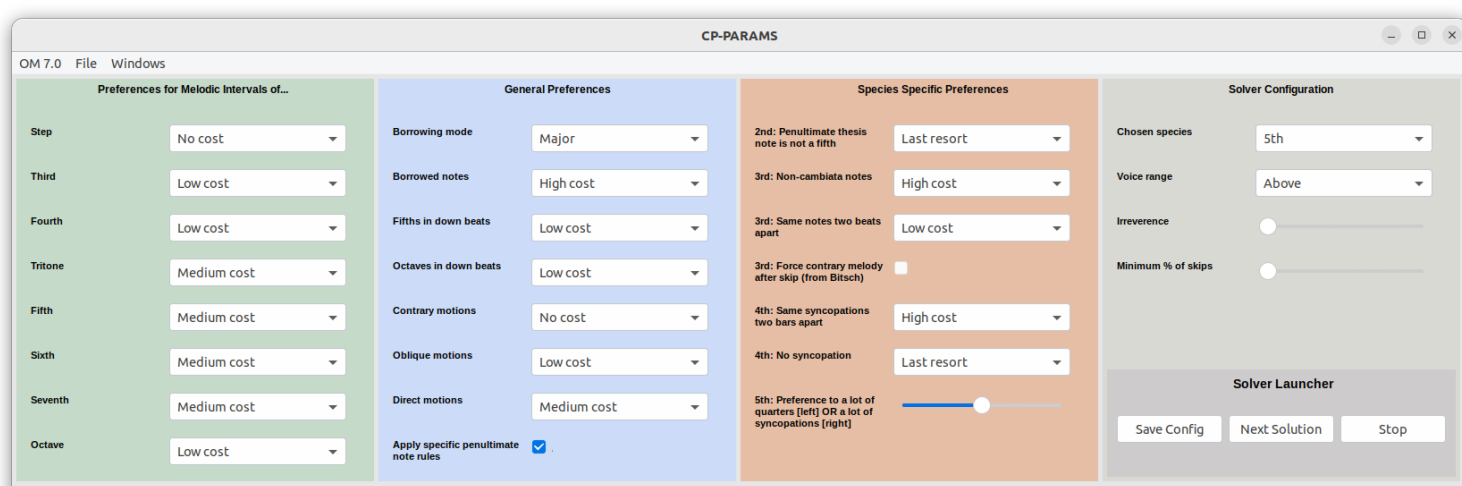


Figure C.4: User interface of the `fuxcp::cp-params` class in OpenMusic.

C.3 Interface Parameters Description

Table C.1 describes all the parameters available in the interface. A low cost represents a high preference while a high cost represents a low preference.

Name	Description	Default value
Step	Preference for melodic intervals of one step or less.	No cost
Third	Preference for melodic third skips.	Low cost
Fourth	Preference for melodic fourth leaps.	Low cost
Tritone	Preference for melodic tritone leaps.	Forbidden
Fifth	Preference for melodic fifth leaps.	Medium cost
Sixth	Preference for melodic sixth leaps.	Medium cost
Seventh	Preference for melodic seventh leaps.	Medium cost
Octave	Preference for melodic octave leaps.	Low cost
Borrowing mode	Type of scale from which notes can be borrowed to generate counterpoint. The first note of the <i>cantus firmus</i> determines the tonic of this scale. Applies everywhere except the penultimate bar.	Major
Borrowed notes	Preference for borrowed notes outside the diatonic scale. These notes are defined by the "Borrowing mode" parameter.	High cost
Fifths in down beats	Preference to have harmonic fifths on the first beat of a bar.	Low cost
Octaves in down beats	Preference to have harmonic octaves on the first beat of a bar.	Low cost
Contrary motions	Preference to have, between two bars, one voice rising while the other is falling.	No cost
Oblique motions	Preference to have, between two bars, one static voice while the other is moving.	Low cost
Direct motions	Preference to have, between two bars, the two voices going in the same direction.	Medium cost
Apply specific penultimate note rules	Force all rules on the notes of the penultimate measure. This mainly refers to the penultimate note that must harmonically be either a major sixth or a minor third depending on whether the counterpoint is above or below.	Checked
2nd: Penultimate thesis note is not a fifth	Preference for the first note of the penultimate bar to be something other than a harmonic fifth	Last resort
3rd: Non-cambiata notes	Preference for the second quarter note of a bar to be a consonance already surrounded by two consonances.	High cost
3rd: Same notes two beats apart	Preference to have the same quarter notes two beats apart. A high cost allows to avoid a certain monotony.	Low cost
3rd: Force joint contrary melody after skip	Force that a melodic skip or leap is followed by a melodic step in the opposite direction.	Unchecked
4th: Same syncopations two bars apart	Preference to have the same half notes two bars apart. A high cost allows to avoid a certain monotony.	High cost
4th: No syncopation	Preference to have distinct half notes instead of syncopations.	Last resort
5th: Preferences to a lot of quarters or a lot of syncopations	Determines the minimum percentage of quarter notes (to the left) and syncopations (to the right) in the fifth species. Pushing the slider all the way to one side is not recommended.	<center>
Chosen species	Determines the type of counterpoint that the tool will generate. From whole notes to syncopations, passing through quarter notes. The fifth species uses the rules and preferences of the previous species.	5th
Voice range	Determines around which pitch the counterpoint will be generated depending on the pitch of the first note of the <i>cantus firmus</i> .	Above
Irreverence	Artificially increases the minimum cost of the solution to obtain counterpoints that are less respectful of the established preferences. Can also be used to get solutions faster.	0
Minimum % of skips	Determines, depending on the counterpoint size, the percentage of melodic intervals larger than one step.	0%
Save Config	Saves all established preferences and allows you to start a new search for this configuration later.	-
Next Solution	Starts or continues searching for the previously saved configuration. Displays a new window with the solution when it is found. Displays an error message if no other solution can be found.	-
Stop	Pause the search. It may take up to 5 seconds.	-

Table C.1: Description of the parameters of fuxcp : : cp-params.