

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DEPARTMENT OF INFORMATION ENGINEERING

OPERATIONS RESEARCH 2

Cplex Installation Guide for Mac OS (ARM) and VSCode

Professor

Fischetti Matteo

Student

Pietrobon Andrea

ACCADEMIC YEAR 2023-2024

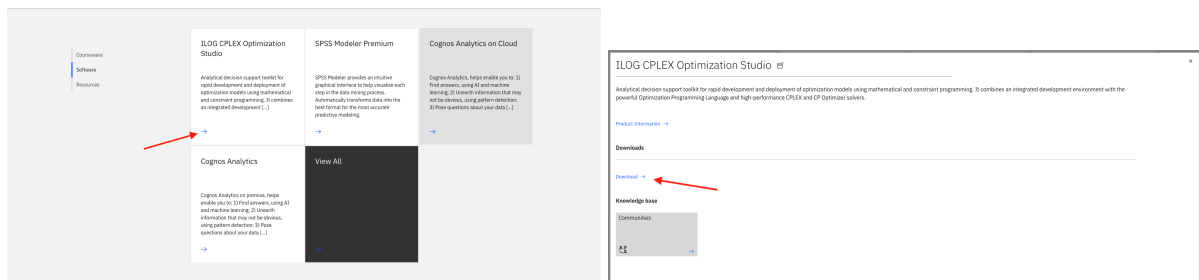
Notes

The following files and an example of their usage are available at the following GitHub link: [TSP_Optimization](#). If there are any mistakes or grammar issues in this file, or if you want to add the steps for installing it in Windows (with and without WSL) and Linux, you can modify it at this link: [Overleaf](#), and send it to the professor again to help the colleagues who will come after you.

Download CPLEX

Visit the website CPLEXonAI and create an account using your university email address. This account will grant you access to download CPLEX.

Once you have completed the account creation process, navigate to the same webpage and locate the **ILOG CPLEX Optimization Studio** section. Click on the arrow next to it, and then click on the **Download** option.



After clicking on the download option, a download screen will appear. If it does not open automatically (typically on Safari), look for the download icon at the top of your browser window and click on it.



From the download screen, select your operating system and architecture and then accept the terms and conditions to initiate the download process.

	View details	License agreement	Download estimate	eAssembly
<input checked="" type="checkbox"/> M08WXML IBM ILOG CPLEX Optimization Studio V22.1.1 for OSX ARM64 09/12/2022 740	View details	License agreement	Download estimate	eAssembly

☒ I agree ☐ I do not agree

By clicking the "I agree" button, you agree that (1) you have had the opportunity to read and understand the above license agreement(s) and multi-product package terms, if any, and (2) terms of the license agreement(s) govern this transaction. If you do not agree with the terms of the agreement(s), you will be unable to download the software.

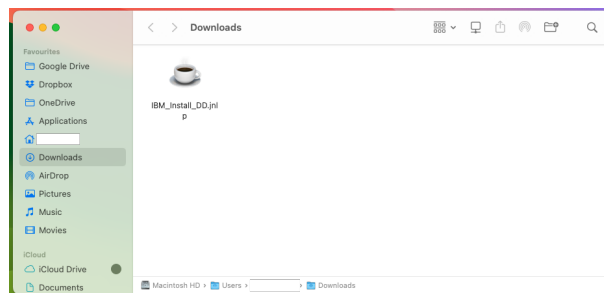
[Download now](#)

If the download does not start or if you encounter a white screen with a red X in the top right corner, close the pop-up and try clicking the download button again. Once the download completes, you will have the CPLEX installer ready for installation on your system.

Note: Before proceeding with the installation, ensure that Java is installed on your machine. If not, you can download and install it from the following link: [Oracle](#).

Install CPLEX

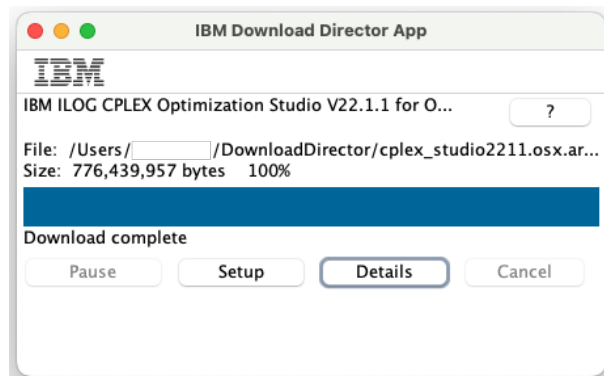
Attempt to open the downloaded file named `IBM_Install_DD.jnlp`.



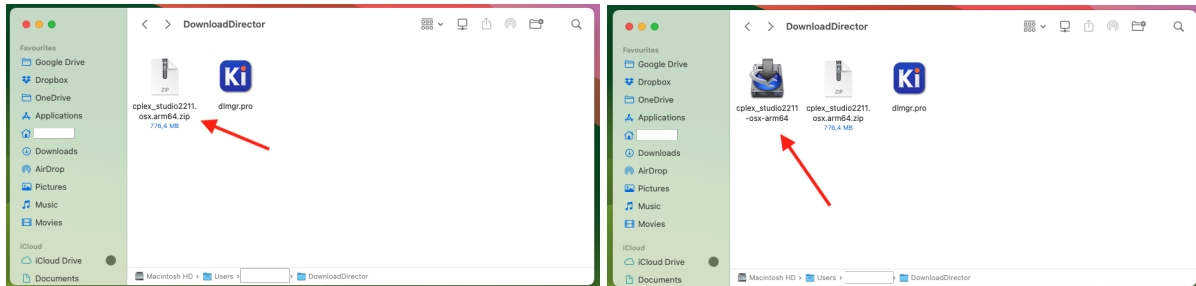
You may encounter an error message on Mac OS indicating that the developer is unidentified. To resolve this issue, follow these steps:

1. Open **System Preferences**.
2. Navigate to **Privacy and Security**.
3. Scroll down to find an option to accept the developer. Click on it to proceed.

Once you have accepted the developer, try opening the downloaded program again. If it doesn't open automatically, locate the downloaded file and click on it. You will be prompted to specify the download location for the CPLEX installer. By default, it is set to `/User/YOUR-NAME/DownloadDirector`. Press Enter to confirm the default location and wait for the download to complete.



After the download finishes, navigate to the DownloadDirector folder and open the downloaded .zip file. Run the CPLEX installation file contained within and follow the installation procedure.



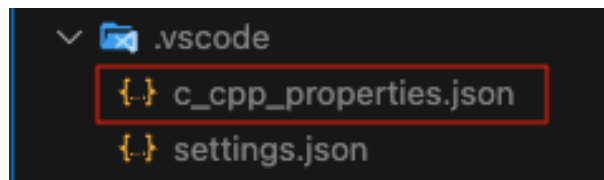
Note: The icons as the one of `dimgr.pro` could be different.

Once the installation is complete, CPLEX and CPLEX Studio will be installed on your machine. You will need to configure the libraries in VSCode to link them appropriately. By default, the installation directory is `"/Applications/CPLEX_Studio2211"`.

Import CPLEX Library on VSCode

The following instructions for using CPLEX in VSCode come with a couple of constraints. Firstly, you must ensure that you have already installed the necessary extensions for writing code in C/C++ and configured CMake to compile your code. Secondly, it's essential to utilize **cl7** as the compiler. If you're uncertain about how to set this up, numerous online guides are available to assist you.

Now that you have all the extensions installed correctly in VSCode, open your current project with VSCode and navigate to the `.vscode` folder. If this folder is not present, you can create it manually (This folder applies the specific configuration settings for this project only and does not affect the entire VSCode configuration). Once inside the `.vscode` folder, open the file `c_cpp_properties.json` (If it does not exist, create it).



If you have created the folder and/or the file **.json** from scratch, copy all the rows in the image (with adjustments for your system on the non-marked rows).

If you already have the folder and the file, just add the two following lines (the ones marked in red) and make sure the others (marked in green) are the same. Ensure that the non-marked paths specified are valid according to the location where you have installed CPLEX Studio.

```
/Applications/CPLEX_Studio2211/cplex/include/**  
/Applications/CPLEX_Studio2211/concert/include/**
```

```

.vscode > {} c_cpp_properties.json > ## version
1
2 {
3   "configurations": [
4     {
5       "name": "Mac",
6       "includePath": [
7         "${workspaceFolder}/**",
8         "/Applications/CPLEX_Studio2211/cplex/include/**",
9         "/Applications/CPLEX_Studio2211/concert/include/**"
10      ],
11      "defines": [],
12      "macFrameworkPath": [
13        "/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/System/Library/Frameworks"
14      ],
15      "compilerPath": "/usr/bin/clang",
16      "cStandard": "c17",
17      "cppStandard": "c++17",
18      "intelliSenseMode": "macos-clang-arm64"
19    }
20  ],
21  "version": 4
22 }

```

Next, navigate to the file CMakeLists.txt and include the following lines just before add_executable. Modify these lines if the path to CPLEX Studio differs:

```

1 cmake_minimum_required(VERSION 3.20)
2 project(TSP_Optimization VERSION 1.0)
3
4 ## find_package(PackageName REQUIRED)
5
6 include_directories(/opt/ibm/ILOG/CPLEX_Studio2211/cplex/include)
7 ## find_package(Boost COMPONENTS filesystem REQUIRED)
8 ## include_directories( ${PackageName_INCLUDE_DIRS} )
9 ## include_directories(${Boost_INCLUDE_DIRS})
10
11
12 # Add the include directory containing vrp.h
13 include_directories(include)
14
15 set(CPACK_PROJECT_NAME ${PROJECT_NAME})
16 set(CPACK_PROJECT_VERSION ${PROJECT_VERSION})
17 set(CMAKE_RUNTIME_OUTPUT_DIRECTORY "${CMAKE_SOURCE_DIR}")
18
19 set(CPLEXDIR /Applications/CPLEX_Studio2211/cplex)
20 set(CMAKE_C_FLAGS "${CMAKE_C_FLAGS} -I. -I${CPLEXDIR}/include/ilcplex")
21 set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -L${CPLEXDIR}/lib/arm64_osx/static_pic -L. -lcplex -lpthread -ldl")
22
23 # Add source files
24 add_executable(TSP_Optimization
25   src/main.c
26   src/algoselector.c
27   src/Algorithm/NN.c
28   src/parser.c
29   src/tsp.c
30   src/Algorithm/optimizations.c
31   src/menu.c
32 )
33
34 # target_link_libraries( ${PROJECT_NAME} ${OpenCV_LIBS} ${Boost_FILESYSTEM_LIBRARY})
35 target_link_libraries(TSP_Optimization PRIVATE ${CPLEX_LIBRARIES} dl m)
36 include(CPack)

```

Finally, as a concluding step, ensure that you include `#include <cplex.h>` in the file where you intend to use CPLEX.

Note: If you are working in a team with another person who has installed CPLEX in a different folder or has another OS, it could be problematic to have everything set in the `CMakeList.txt` file. For this reason, it's better if you comment out the first line that you have added:

```
set(CPLEXDIR /Applications/CPLEX_Studio2211/cplex)
```

For every build, you can manually add the parameter with `cmake`:

```
cmake -DCPLEXDIR="/Applications/CPLEX_Studio2211/cplex"
```

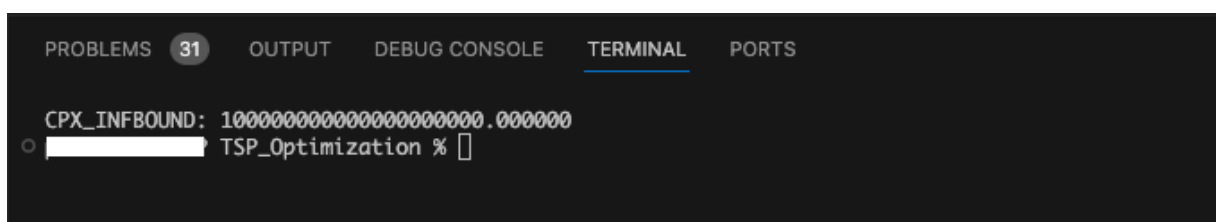
(not necessary for the make).

Functionality Testing

At this stage, it's time to test the functionality of CPLEX. To verify that everything is installed and configured correctly, add the following lines to your main file and run the program.

```
src > C main.c > ...
1
2 #include <stdio.h>
3 #include <cplex.h>
4
5 int main(int argc, char** argv)
6 {
7     // Print CPX_INFBOUND
8     printf("CPX_INFBOUND: %f\n", CPX_INFBOUND);
9 }
10
```

If everything has been set up correctly, the program will compile without errors, and it will print the value on the terminal as shown in the screenshot below.



```
PROBLEMS 31 OUTPUT DEBUG CONSOLE TERMINAL PORTS
CPX_INFBOUND: 10000000000000000000.000000
TSP_Optimization %
```

You can now remove the test lines and proceed with your project by implementing various functions and methods of CPLEX.

Note: The following files and an example of their usage are available at the following GitHub link: https://github.com/Piero24/TSP_Optimization. If there are any mistakes or grammar issues in this file, or if you want to add the steps for installing it in Windows (with and without WSL) and Linux, you can modify it at this link: <https://www.overleaf.com/5545632258hqfpqnvbhsfr#df0b00>, and send it to the professor again to help the colleagues who will come after you.