Appendix A

Development environment

This appendix is dedicated to deepen the integration of external software to the project's source code in a Microsoft Windows environment. The first section offers a guide on how to find the static C library provided by CPLEX and how to integrate it with Microsoft Visual Studio IDE, so that you can use all the CPLEX functions in the source code. The second section instead shows the use of Gnuplot, a software capable of creating graphs and which will be used to give also a graphical representation of the TSP solution found by the code.

A.1 CPLEX library integration with Visual Studio

The first section shows the integration of CPLEX with Microsoft Visual Studio IDE. In details, we will provide a guideline on how the link of the external library `cplex.h` to the source code of the project.

This guide uses the following software:

- Microsoft Visual Studio Community (version 16.10.0)
- IBM ILOG CPLEX Optimization Studio (version 20.1.0)

but the steps are general and therefore easily replicable with other different versions.

First of all, we have to download and install IBM ILOG CPLEX Optimization Studio. The software is usually released after having paid the relative license, but thanks to an agreement between IBM and the University of Padua, students are entitled to have a free license which can be used only and exclusively for educational purposes. Hence, after you have successfully identified yourself as a university student on the IBM site [10], you can get the installer for free.

Following the hints provided by the installation process, an optimization software that solves mixed integer linear (or quadratic convex) programming problems, even of considerable size, will be installed. The program can be started through a dedicated graphical interface or from the command line, but for our purposes, the most important thing is that it is made available the static C library, which we will later connect with the IDE and then use it in the source code.

After that, we have to download and install Microsoft Visual Studio Community. Visual Studio is an IDE (*Integrated Development Environment*), that is, a software capable of designing, compiling and running applications in various programming languages, including C and C ++. During the installation process you need to choose the Microsoft .NET framework installation package, in order to install the right C and C ++ compiler.

The Community version, unlike the Professional and Enterprise versions, is totally free and, after an initial 30-days trial, you can continue to use the IDE without any restrictions by logging in with your Microsoft account or by creating a new one at the moment.

At this point, the two necessary software are correctly installed and the next step will be to link them together. In the end the result will be to have an IDE able to develop C applications with all the default libraries, in addition to the `cplex.h` library, used to solve mathematical programming problems.

To create a new project, open Visual Studio, select "Create a new project" and choose an empty project. After setting the name and the path where to save the project, the development environment will appear as in Figure A.1.

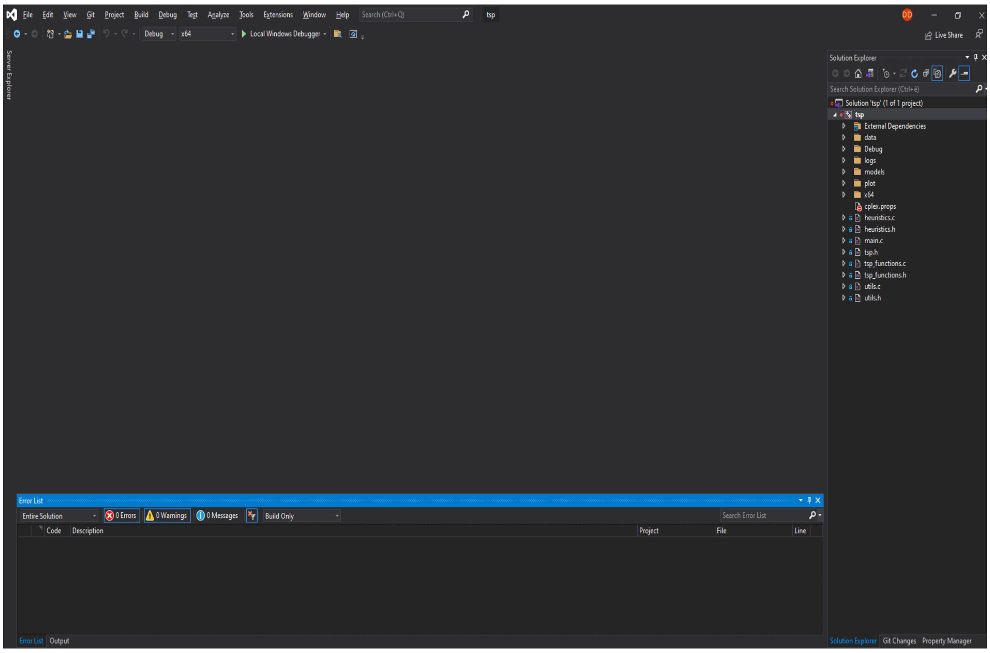
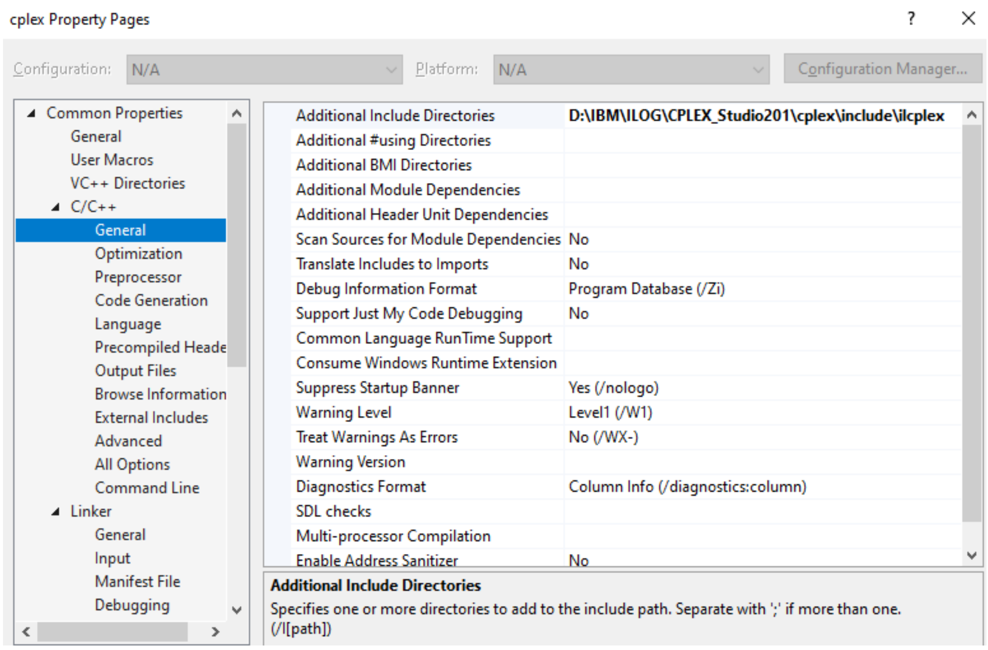


Figure A.1: Visual Studio environment. On the right you can see the "Solution Explorer" tab, where you can manage all the files belonging to the project, and some other customized tabs.

To use CPLEX within the newly created project it is necessary to perform some operations to indicate to the compiler the position of the CPLEX header files and to the linker the position of the relative libraries. To do this, select the "Project" tab and then select "*project_name* properties". In the window that opens, choose "Configuration Manager" and select New from "Active solution configuration" and x64 from "Active solution platform", hence save the changes and close the windows.

Then, open the tab "View" > "Other Windows" > "Property Manager", hence right click on the project name, select "Add New Project Properties Sheet" and choose a name and a path, then press "Add". This property sheet, once configured with all the references to the header files and the CPLEX libraries can be easily added to other projects.

After that, in the "Property Manager" tab, located to the right of the screen next to the "Solution Explorer" and "GIT Changes" tabs, expand the project name, then expand "Release | 64" and double click on the previously created property window. Figure A.2 shows the window that will appear on the screen, where the following fields must be entered:



Additional Include Directories	D:\IBM\ILOG\CPLEX_Studio201\cplex\include\ilcplex
Additional #using Directories	
Additional BMI Directories	
Additional Module Dependencies	
Additional Header Unit Dependencies	
Scan Sources for Module Dependencies	No
Translate Includes to Imports	No
Debug Information Format	Program Database (/Zi)
Support Just My Code Debugging	No
Common Language RunTime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	Level1 (/W1)
Treat Warnings As Errors	No (/WX-)
Warning Version	
Diagnostics Format	Column Info (/diagnostics:column)
SDL checks	
Multi-processor Compilation	
Enable Address Sanitizer	No

Additional Include Directories
Specifies one or more directories to add to the include path. Separate with ';' if more than one. (/I[path])

Figure A.2: Visual Studio project property window.

- In C / C ++ > General > Additional Include Directories:

  Add the path to the directory with the CPLEX files to include (usually the path is like `D:\IBM\ILOG\CPLEX_Studio201\cplex\include\ilcplex`).

- In C / C ++ > Preprocessor > Preprocessor Definitions:

  Add IL_STD.

- In C / C ++ > Advanced > Calling Convention:

  Choose __stdcall (\Gz).

- In C / C ++ > Advanced > Compile As:

  Choose Compile as C code (\TC).

- In C / C ++ > Advanced > Disable Specific Warnings:

  Add 4996.

- In Linker > General > Additional Libraries Directory:

  Add the the path to the directories containing the libraries compiled for the correct version of Visual Studio (usually the path is like `D:\IBM\ILOG\CPLEX_Studio201\cplex\lib\x64_windows_msvc14\stat_mda`).

- In Linker > Input > Additional Dependencies:

  Add `cplex2010.lib;ilocplex.lib`.

Once all the aforementioned fields have been entered, press "Apply" and exit. From now on, in order to use any CPLEX function, just include the header `cplex.h` in the source files.

A.2 Gnuplot

Gnuplot is a portable command-line driven graphing utility for Linux, OS/2, MS Windows, OSX, VMS, and many other platforms. The source code is copyrighted but it is freely distributed. It was originally created to allow scientists and students to visualize mathematical functions, but has grown to support many non-interactive uses such as web scripting. The software has been supported and under active development since 1986.

Gnuplot supports many different types of 2D and 3D plots, as well as many different types of output modality: interactive screen display or direct output to file, including the most important formats such as pdf, png, gif, jpeg and also LaTeX.

The current version is 5.4 (last release 5.4.1), while the development version is 5.5. The current version is downloadable from the *Gnuplot* [9] official web site, where there are also the documentation, the link to the *Git* repository of the development version and other useful material.

The first step to perform is installing the program, following the hints provided during the installation process. Once the software is installed, it can be used both using a graphical tool or as a series of commands in the shell. For our purposes, we have decided to follow the second way.

After that, we have to integrate Gnuplot inside the source code, and the strategy used to plot the solution is the following:

1. During the scan of the `best_sol` field of the `struct` *instance*, if the edge is selected by the solution, write a file with the coordinates of that edge (i.e., the x and y coordinates of the two points that build the edge) followed by a double CR LF. The operations of opening and writing on the file are performed using the classic functions made available by the `stdio` library.

In this way, Gnuplot will plot several small independent graphs, each of them representing one edge of the entire TSP solution. Once the scan is finished, the resulting file will have the following format:

```
x[i]  y[i]
x[j]  y[j]


x[i]  y[i]
x[j]  y[j]


 ...    ...
 ...    ...
```

2. At the end of the writing phase, use the `system()` function to call the related `gnuplot` command in order to start the actual plot of the solution:

```
system("gnuplot ./plot/commands.txt");
```

where the unique parameter is a path to the `commands.txt` file which contains the set of instructions in order to to build the graph and plot the coordinates file created previously. This file has the following content:

```
set style line 1 \
    linecolor rgb '#FF0000' \
    linetype 1 linewidth 2 \
    pointtype 7 pointsize 2


plot "./plot/data.dat" with linespoints linestyle 1
pause mouse close
```

where first of all, it set the style of the plot such as the color and the type of line/point to use. Then it start the real plotting phase with the `plot` command, where it is specified the path to the coordinates file, the previous created style and the plot title.

The overall result of this simple integration is the visualization on the screen of a graph representing the points and the TSP selected edges for the instance provided in input by the user, as we can see in Figure A.3. As we can see from the figure, Gnuplot offers the possibility to export the obtained graph in PDF and SVG format or save it as an image. Furthermore, there are buttons to show or not the grid, to go through the various actual zooms, as well as to make an autoscale and return to the default resolution. Finally, there is also the possibility to access the terminal configuration for greater customization of the overall environment.

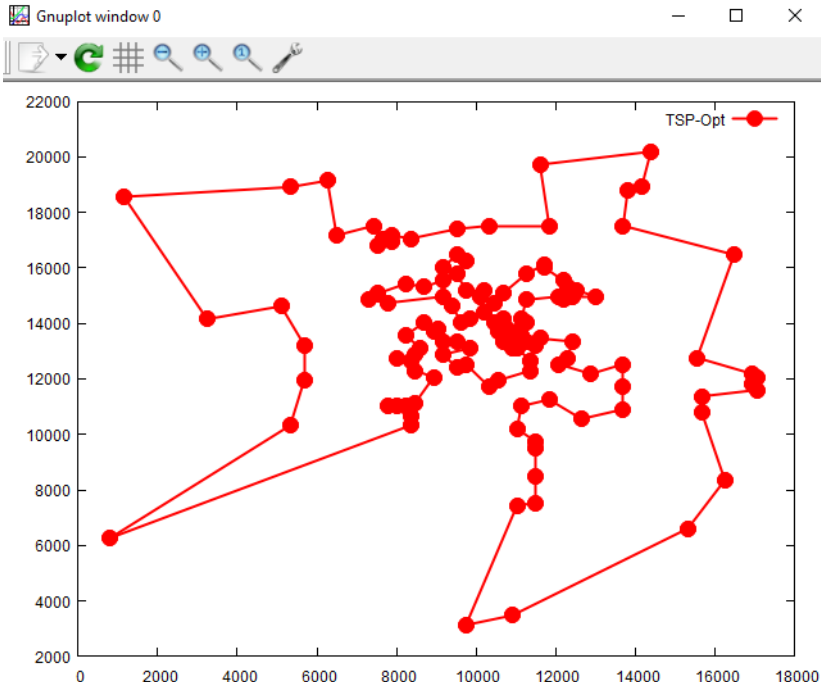


Figure A.3: Plot of the final result returned by the application (file bier127.tsp)