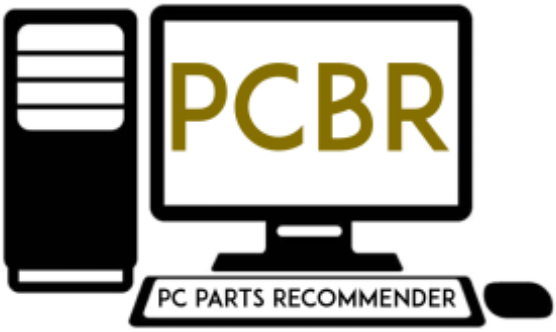

User Manual

Version 1.0.0

Víctor Badenas Crespo
Michael Birkholz
Andrea Masi
Kevin David Rosales Santana

June 16, 2021

Contents

1 Introduction

This is the user guide for the PCBR recommender. There are two ways of using the software, through a command line interface (CLI) or with a graphical user interface (GUI). Both of the methods require an equivalent setup.

2 Directory Structure

From the root folder, the **data** folder contains the initial case base library, additional information tables for the components and the feature scalers' and normalizers' parameters in the file **feature_scalers.json**. Additionally, the **feature_relevance.csv** matrix is located in that folder. This file contains the contribution of each of the user's preferences to each of the features used by the system.

The **interface** directory contains the PyQt5 files for running the application's GUI. The **src** folder contains all the code required for the core PCBR application to run in an interactive shell as well as the code underlying the algorithms used for both the CLI and the GUI. Finally, the **test** folder contains unit test files.

3 Setup

The application was developed in a **python3.6** environment. The required packages can be found in the **requirements.txt** file located in the root directory of the software package. A conda environment is highly encouraged. First create and activate the environment with the commands in Listing 1 and install the required packages using the commands in Listing 2. The commands in the latter listing will install all of the dependencies.

```
conda create --name pcbr python=3.6
conda activate pcbr
```

Listing 1: Instructions to create the conda environment

```
python -m pip install --upgrade pip
pip install -r requirements.txt
```

Listing 2: Instructions to install dependencies in environment

4 GUI

The GUI for the project can be run from the root directory of the software with the following command:

```
python interface/welcomeWindow.py
```

Note that running the GUI requires a shell with display capabilities and the package PyQt5, previously installed according to the instructions in Listing 2. The *welcomeWindow.py* file additionally allows the user to specify an alternative CBL file in csv format, similar to *data/pc_specs.csv*, using the "-c" option.

When the start button is pressed, the user is prompted with a window with 3 tabs and a **Done** button. The tabs correspond to the **Basic**, **Preferences** and **Advanced** as

shown in 2, 3 and 4, respectively. In the figures, we show the empty questionnaire on the left and an example of a filled-in questionnaire on the right. The Done button will be inactive until all the fields have been filled in correctly.

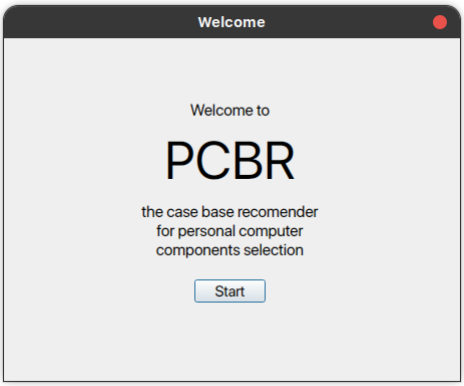

Figure 1: Welcome window

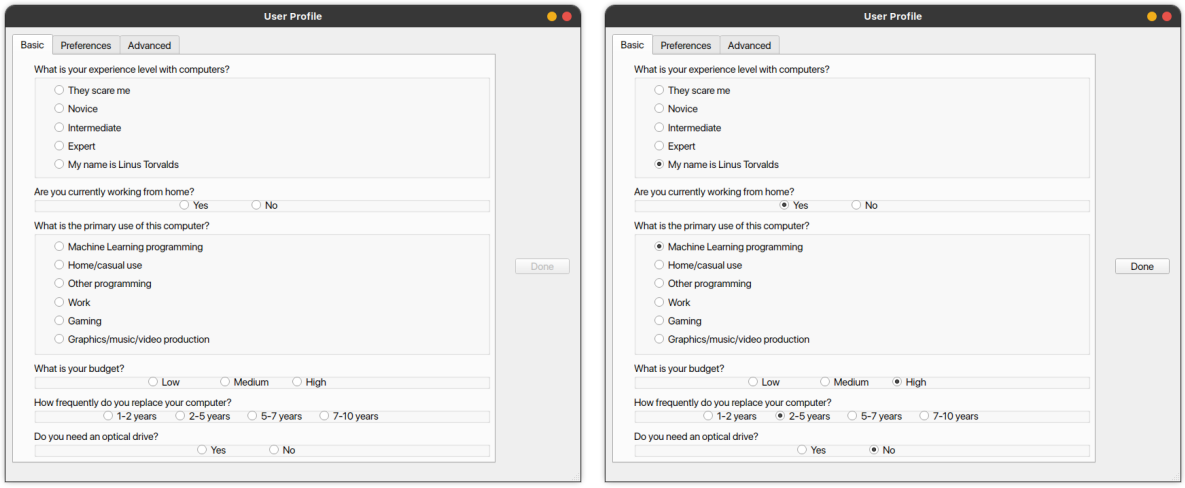

Figure 2: Empty and full Basic page

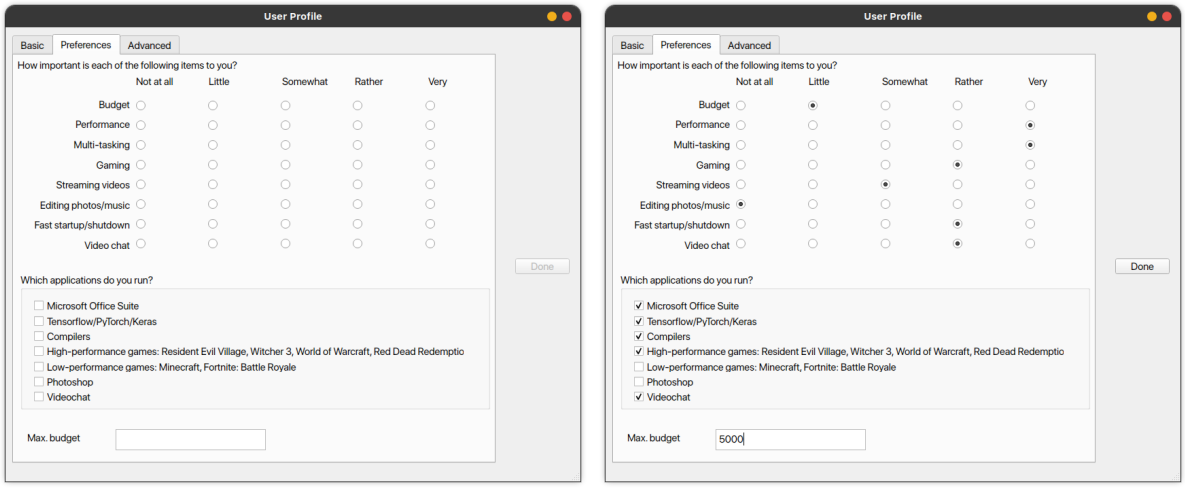

Figure 3: Empty and full Preferences page

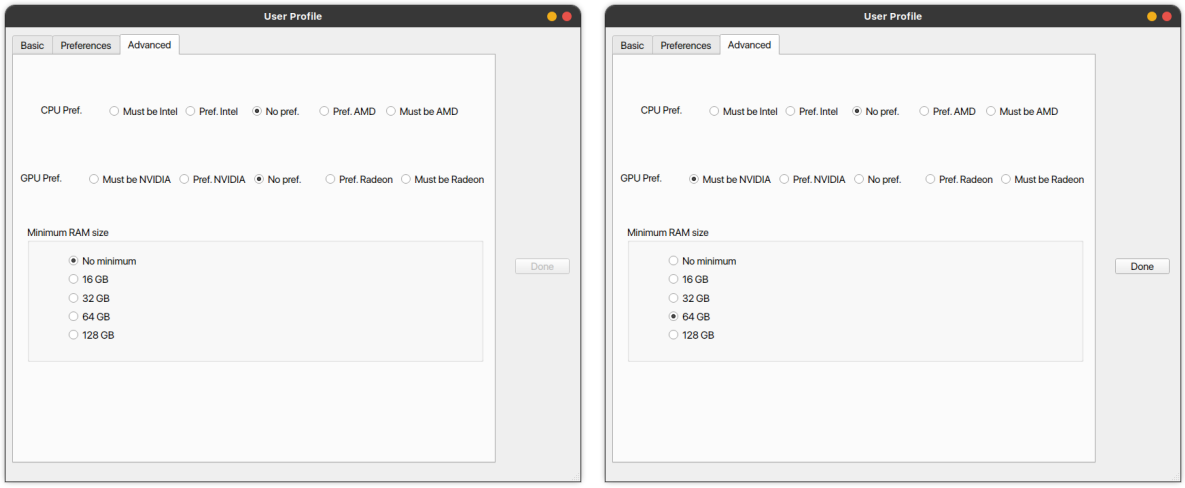

Figure 4: Default and full Advanced page

After the user has filled in the questionnaire and pressed the done button, they are prompted with a solution. The user can decide if it is acceptable or not and decide if they want to change any component as shown in Figure 5.

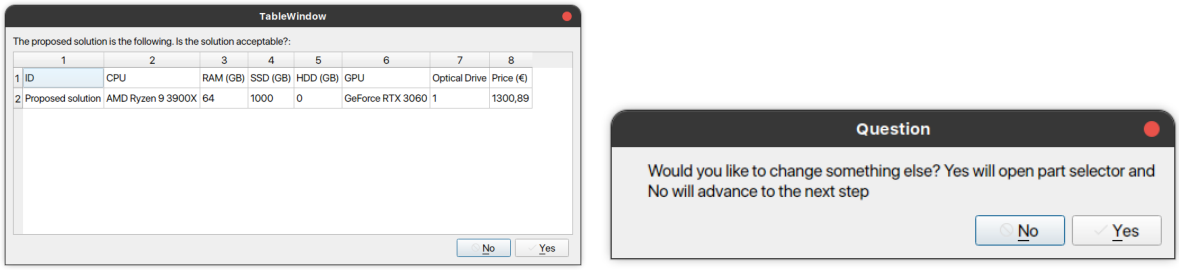

Figure 5: Proposed solution and question window

Then, the user is prompted with a window to choose the component to change and immediately after, the component that they would like to substitute it with. This process is shown in Figure 6.

After the user has selected the component to be replaced, they are prompted with all the solutions up to this point and asked if they would like to change anything else. If the user selects yes, they are taken back to the component selection stage (where they will not see the components that have already been modified). If the user selects no or all components have been changed, the user is asked to choose between the proposed solutions as shown in Figure 7.

Finally the user is shown the selected instance 8 and returned to the main questionnaire window.

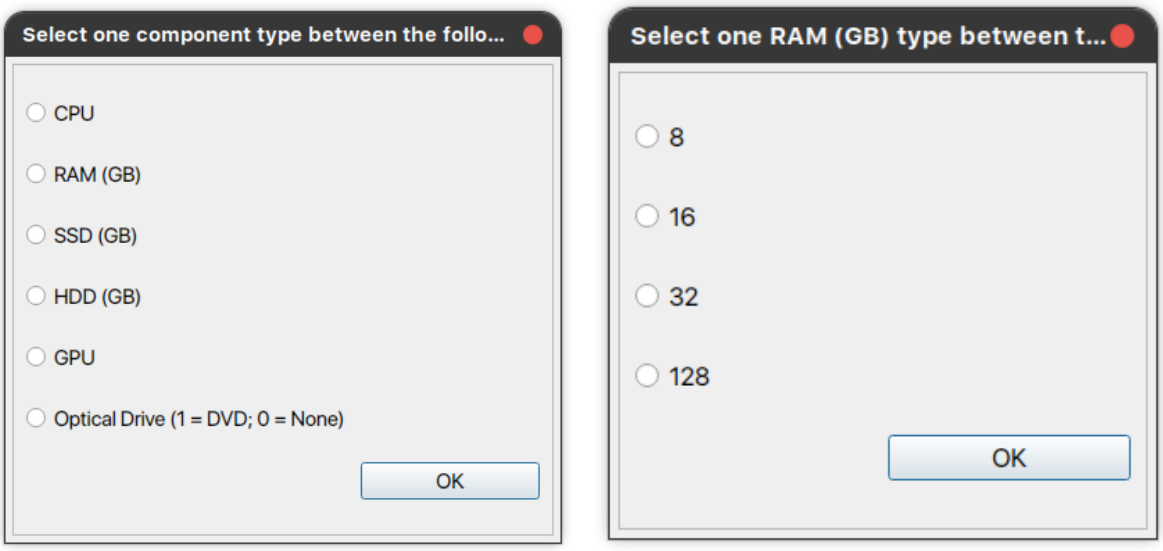

Figure 6: Component type and value selection

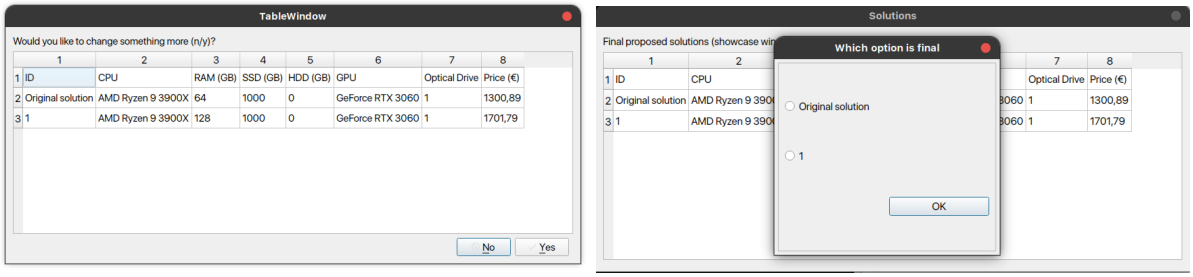

Figure 7: Change something else and final selection

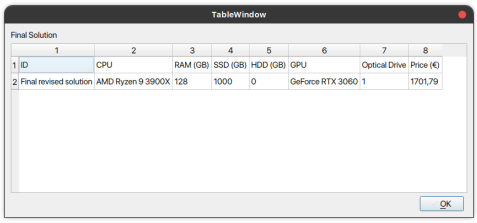

Figure 8: Final showcase

5 CLI

To run the software through CLI[1], the following command has to be run from the **src** directory:

```
python pcbr.py
```

The *pcbr.py* file additionally allows the user to specify an alternative CBL file in csv format, similar to *data/pc_specs.csv*, using the "-c" option. The "-g" option will generate 10000 random entries into a csv in *data/generator.csv* to evaluate the time response of the system as well as the evolution of the number of retained instances. Once the program has been started, a prompt will come up asking the user to input a series of

[1]While possible to run through the CLI, the team recommends running through the GUI for the best, most intuitive experience

values as shown in Figure 9. At this prompt, or any of the 3 data input stages, the string **exit** can be used to exit the program. The input for this first prompt must be in the order of Table 1, separated by commas:

position	name	type and range
1	Experience	int $\in \{1, 2, 3, 4\}$
2	Work from Home status	int $\in \{0, 1\}$
3	Primary use	string $\in$ {Home, Gaming, Production, Programming, ML, Work}
4	Budget	int $\in \{1, 2, 3\}$
5	Replacement frequency	int $\in \{1, 2, 3, 4\}$
6	Office	int $\in \{0, 1\}$
7	Photoshop	int $\in \{0, 1\}$
8	Video Chat	int $\in \{0, 1\}$
9	ML	int $\in \{0, 1\}$
10	Compilers	int $\in \{0, 1\}$
11	High-Performance Games	int $\in \{0, 1\}$
12	Low-Performance Games	int $\in \{0, 1\}$

Table 1: Table of user profile features

Figure 9: Profile input prompt

Once the user profile has been entered into the system, the preferences are requested following a similar format to the user profile. The prompt appears as shown in Figure 10. The values entered should reflect the importance of a given feature to the user and should be entered in the order in Table 2.

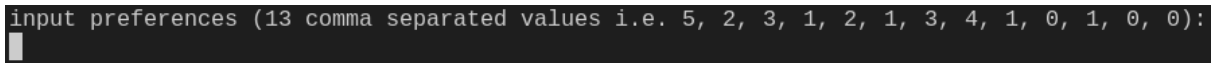

Figure 10: Preferences input prompt

Once the user preferences are entered, the user is prompted for constraints, as shown in Figure 11. The values must be entered in a key:value pair, comma-separated string as shown in the example.

Figure 11: Constraints input prompt

After all the data has been entered, the user is prompted with a proposed solution and asked if the solution is satisfactory or not as shown in Figure 12. If the user considers that the solution is satisfactory, the program will skip to the step which displays the final revised solution, as depicted in Figure 18. If the user does not consider the solution good enough, they are directed to the next step.

If the user decides that the proposed solution is not satisfactory, the program will ask if the user would like to change any component of the configuration as shown in Figure 13.

position	name	type and range
1	Budget	int $\in \{1, 2, 3, 4, 5\}$
2	Performance	int $\in \{1, 2, 3, 4, 5\}$
3	Multi-tasking	int $\in \{1, 2, 3, 4, 5\}$
4	Gaming	int $\in \{1, 2, 3, 4, 5\}$
5	Streaming videos	int $\in \{1, 2, 3, 4, 5\}$
6	Editing videos/photos/music	int $\in \{1, 2, 3, 4, 5\}$
7	Fast startup/shutdown time	int $\in \{1, 2, 3, 4, 5\}$
8	Video chat	int $\in \{0, 1\}$
9	Microsoft Office Suite	int $\in \{0, 1\}$
10	Tensorflow, PyTorch, Keras	int $\in \{0, 1\}$
11	Compilers	int $\in \{0, 1\}$
12	High-performance games	int $\in \{0, 1\}$
13	Low-performance games	int $\in \{0, 1\}$

Table 2: Table of preferences

Figure 12: Proposed solution prompt

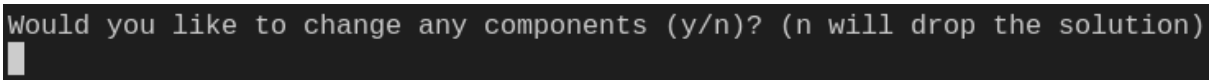

Figure 13: Component question prompt

If the user chooses to change a component, the program asks which one they would like to change as shown in Figure 14. The selection is performed with a numeric index input. If a component has already been changed, it will not appear in this list.

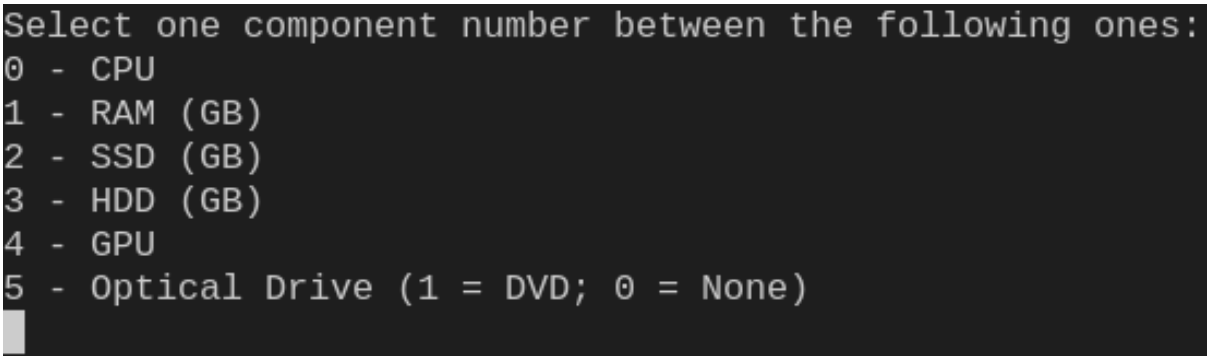

Figure 14: Component selection prompt

Once the component has been selected, the user is asked which value they would like to use instead of the one previously selected, as can be observed in Figure 15. Selection is carried out using a numeric integer value from the given list.

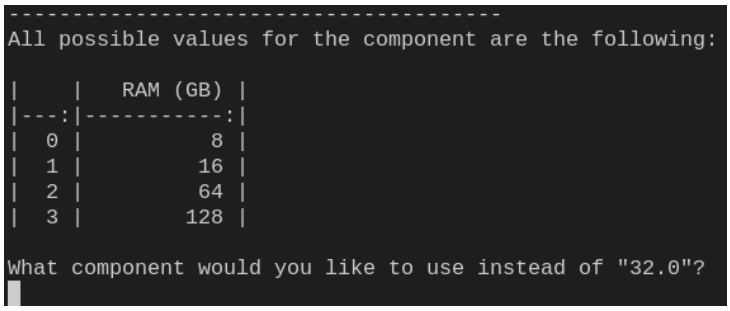

Figure 15: Component value selection prompt

Once the component has been chosen, the user is shown all of the partially-modified solutions up to this point and is asked if they would like to change another component as shown in Figure 16. If the user chooses to change another component, they are prompted with the component question again. If the user chooses to not change anything else, they are prompted with the final selection stage as in Figure 17 to select which of the solutions they would like to keep.

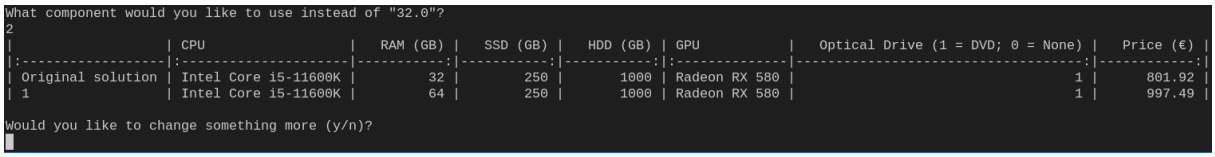

Figure 16: All solutions display prompt

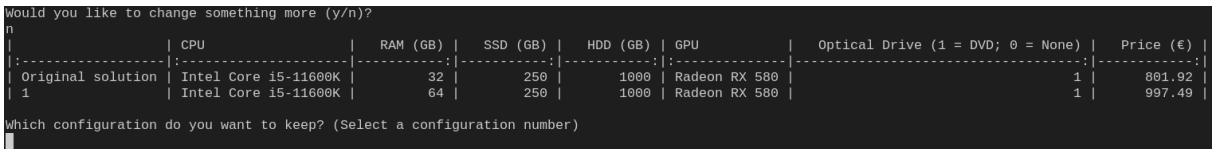

Figure 17: Final selection prompt

Finally, the revised solution is shown to the user along with info about the retention of the case. This prompt is shown in 18.

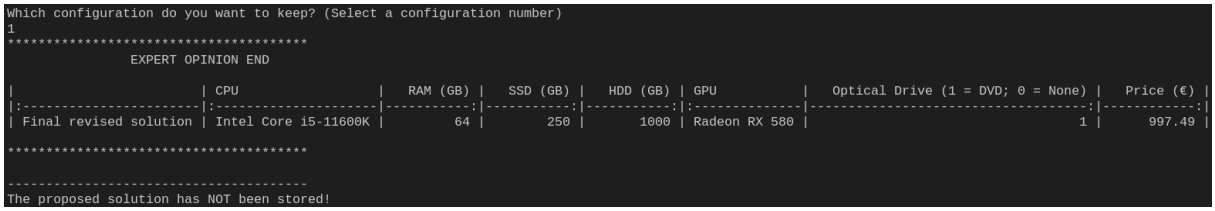

Figure 18: Final revised solution display