Installation

UNIX/Linux

```
unzip pamilo.zip
cd pamilo
cmake .
make
```

Windows

This is how I've got pamilo to run:

- Unzip pamilo.zip and go into the directory it is unzipped into.
- Run `<path to cmake>\cmake.exe .` . If you have a standard installation of CPLEX this should suffice, but sometimes it does not. Refer to the CMake section below. I have not testet CDD on Windows.
- Start Visual Studio and build pamilo_cli in Release mode.
- Go to `<path to pamilo>\Release` and run `pamilo_cli.exe` (for detailed usage see below).

As I am no expert in Windows whatsoever, please feel free to let me know if the configuration and compilation process can be made more efficient or the descriptions more precise.

CMake

Version 3.18 or higher is needed.

CPLEX

Version 12.9 or higher is needed, as we rely on the multiobjective optimization features introduced in in 12.9.

If CPLEX is installed in a default location (`/opt/ibm/ILOG/CPLEX_Studio12<9-10>` or `C:/Program Files/IBM/ILOG/CPLEX_Studio12<9-10>`), it should be found automaticly. If it is installed in such a way that the directories `<path to cplex>/concert` and `<path to cplex>/cplex` are present you can set CPLEX to the `<path to cplex>`. Otherwise you have to set `CONCERT_INCLUDE_DIR` and `CPLEX_INCLUDE_DIR` to the directories containing `ilconcert/iloenv.h` and `ilcplex/ilocplex.h` respectively and `CONCERT_LIB`, `CPLEX_LIB`, and `ILOCPLEX_LIB` to the corresponding static libraries. Furthermore, on UNIX `CPLEX` needs the libraries `dl` and `pthread`.

CDD

For four or more objectives the vertex enumeration can be sped up by using the `cddlib`. This is done by activating the CMake flag `USE_CDD` and (if they are not found automatically) providing `libcdd.a` in `CDD_LIB` and the directory containing `cdd.h` in `CDD_INCLUDE_PATH`. The cddlib can be found at https://github.com/cddlib/cddlib.

Usage

`<pamilo_cli> = ./pamilo_cli` on UNIX and `<pamilo_cli> = Release\pamilo_cli.exe` on Windows.

```
<pamilo_cli> [<parameters>] <instance>
```

Important parameters are:

`-o <output>` The basename for all output files. This defaults to on UNIX and an empty string on Windows.

`-e <epsilon>` Epsilon to be used in floating point calculations.

`<pamilo_cli> -h` shows all available parameters.

Instance format

An instance is to be in CPLEX `.lp` format with multiple objective functions. All multiobjective parameters (Priority, Weight, AbsTol, and RelTol) are ignored. If the demand is there this could be changed for the tolerances.

Output

A list of all extreme points is printed to the command line. Every line is one point, where the coordinates are separated by spaces. `<output>_log` logs the running time, the vertex enumeration time and the time that CPLEX took. If you want anything else to be logged, let me know. `<output>_sol` stores all extreme points with their corresponding variable assignment. Every line has the form:

```
[ <obj1> <obj2> ... ] <var1>=<val1> <var2>=<val2> ...
```

This format is the one used by PolySCIP. If you prefer any other solution format (e.g. xml or json based, like CPLEX's .sol), let us know.