clpAPI – Quick Start

Gabriel Gelius-Dietrich

February 13, 2020

1 Introduction

The package *clpAPI* provides a low level interface to the C API of COIN-OR Clp[1] (COIN-OR linear programming). The package *clpAPI* relies on a separate installation of COIN-OR Clp.

2 Installation

See `INSTALL` for installation instructions and platform specific details.

3 Usage

In the following, an example lp-problem will be created and solved:

maximize

$$z = 5x_1 + 4x_2 + 3x_3$$

subject to

$$\begin{aligned} 2x_1 + 3x_2 + \ x_3 \ &\leq 5 \\ 4x_1 + \ x_2 + 2x_3 &\leq 11 \\ 3x_1 + 4x_2 + 2x_3 \ &\leq 8 \end{aligned}$$

where all variables are non-negative

$$x_1 \geq 0, \ x_2 \geq 0, \ x_3 \geq 0$$

Load the library.

```
> library(clpAPI)
```

Create a problem object.

```
> prob <- initProbCLP()
```

Set the direction of optimization (-1: maximize, 1: minimize).

[1] COIN-OR linear programming version 1.12.0 or higher `https://projects.coin-or.org/Clp`

```
> setObjDirCLP(prob, -1)
```

Prepare data structures for the problem object. Number of columns and rows:

```
> nc <- 3
> nr <- 3
```

The constraint matrix is passed in column major order format. **Be careful here:** all indices start with 0! Row indices.

```
> ia <- c(0, 1, 2, 0, 1, 2, 0, 1, 2)
```

Column indices.

```
> ja <- c(0, 3, 6, 9)
```

Non-zero elements.

```
> ar <- c(2, 4, 3, 3, 1, 4, 1, 2, 2)
```

Lower bounds for the variables (columns).

```
> clb <- rep(0, 3)
```

Right hand side (row upper bounds for the rows).

```
> rub <- c(5, 11, 8)
```

Objective coefficients.

```
> obj <- c(5, 4, 3)
```

Load problem data into the problem object.

```
> loadProblemCLP(prob, nc, nr, ia, ja, ar, clb, NULL, obj, NULL, rub)
```

Solve the problem using the simplex algorithm.

```
> solveInitialCLP(prob)
```

```
[1] 0
```

Retrieve the value of the objective function after optimization.

```
> getObjValCLP(prob)
```

```
[1] 13
```

Retrieve the primal values of the structural variables (columns) after optimization.

```
> getColPrimCLP(prob)
```

```
[1] 2 0 1
```

Retrieve the dual values of the structural variables (columns) after optimization (reduced costs).

```
> getColDualCLP(prob)
```

```
[1]  0 -3  0
```

Free memory, allacated to the problem object.

```
> delProbCLP(prob)
```

4 Function names

4.1 Searching

The function names in *clpAPI* are different from the names in COIN-OR Clp, e. g. the function `addColsCLP` in *clpAPI* is called `Clp_addColumns` in COIN-OR Clp. The directory `inst/` containes a file `c2r.map` which maps a COIN-OR Clp function name to the corresponding *clpAPI* function name. Additionally, all man-pages contain an alias to the COIN-OR Clp function name. The call

```
> help("Clp_addColumns")
```

will bring up the man-page of `addColsCLP`.

4.2 Mapping

The file `c2r.map` in `inst/` maps the *clpAPI* function names to the orininal COIN-OR Clp function names of its C-API. To use the latter, run

```
> c2r <- system.file(package = "clpAPI", "c2r.map")
> source(c2r)
```

now either

```
> pr1 <- initProbCLP()
> delProbCLP(pr1)
```

or the original functions

```
> pr2 <- Clp_newModel()
> Clp_deleteModel(pr2)
```

work both. Keep in mind that the mapping only affects the function names not the arguments of a function.