Optidef
A Latex library for optimization problems
Version - 3.1

Jesus Lago

April 23, 2019

Contents

1 Introduction and features

This Latex library provides a standard set of environments for writing optimization problems. The most important features are:

1. It references optimization problem using three different policies: no equation is referenced, the problem is referenced with a single label, each equation has an individual reference. For more details refer to Sections 3 and 4.

2. It defines two problem size formats: a long format and a short format. For more details refer to Sections 3 and 5.

3. It allows four different outputs for the location of the constraints. For more details refer to Sections 3 and 6.

4. It allows the definition of a limitless number of constraints. For more details refer to Section 3.2.

5. Four different type of problems: *minimize*, *maximize*, *arg min* and *arg max*. For more details refer to Sections 3 and 4.

6. The optimization problem can be broken in several pages without compromising the alignment or the structure of the problem. For more details refer to Section 7.

7. The objective function can be broken in several lines without compromising the alignment or the structure of the problem. For more details refer to Section 8.

2 Using the package

The package can be imported by directly adding

```
\usepackage{optidef}
```

to the document preamble. When importing the packages three options can be used, `short`, `nocomma`, and either `c1`, `c2`, or `c3`:

```
\usepackage[short,c1|c2|c3,nocomma]{optidef}
```

The first option changes the default long format of the optimization problems to a shorter format; for a better explanation (including examples) of the `short` option check Section 5.

The options `c1`, `c2`, and `c3` change the default format of the constraints; the default format is format 0 (as defined in Section 6); `c1`, `c2`, and `c3` respectively change the default constraint arrangement to format 1, 2, and 3. For a better explanation of the four formats including examples, we refer to Section 6.

For the `nocomma` option check Section 9. For a detailed description of how to use the package keep reading the next section.

3 Environment Syntax Definition

Considering that `Const.i` stands for constraint i, `LHS.i` stands for the left-hand-side of constraint i, and `RHS.i` for the right-hand-side counterpart, the basic structure to define a general optimization problem with N constraints is:

```
\begin{mini#}|sizeFormat|[constraintFormat]<break>
{optimizationVariable}
{objectiveFunction\label{objective}}
{\label{optimizationProblem}}
{optimizationResult}

\addConstraint{LHS.1}{RHS.1\label{Const1}}{extraConst1}
\addConstraint{LHS.2}{RHS.2\label{Const2}}{extraConst2}
.
.
\addConstraint{LHS.N}{RHS.N\label{ConstN}}{extraConstN}
\end{mini#}
```

3.1 Definition of Problem parameters

(i) `mini#`: defines the type of environment and reference used. There are four environments: `mini`, `maxi`, `argmini`, and `argmaxi`. There are three types of referencing: `mini`, `mini*` and `mini!`. Consult Section 4 for more details.

(ii) (Optional) `sizeFormat`: optional parameter to define the size format of the problem. The possible values are:

- l: for the long format as defined in Section 5.
- s: for the short format as defined in Section 5.

(iii) (Optional) `constraintFormat`: optional parameter to change the format of the constraints. The parameter `constraintFormat` can take the following values:

- 0: for the Standard definition in Section 6.
- 1: for Alternative 1 in Section 6.
- 2: for Alternative 2 in Section 6
- 3: for Alternative 3 in Section 6

(iv) (Optional) `break`: optional parameter to allow the optimization problem to break across multiple pages. For details on this feature, check Section 7.

(v) `optimizationVariable`: variable to be optimizated in the problem, e.g. $w \in \Re^N$.

(vi) `objectiveFunction\label{objective}`: function to be minimized/maximized as a function of the optimization variable, e.g. $\|w\|_2$. If required, the objective function label should also be included withing this term

(vii) `\label{optimizationProblem}`: it defines the main and general reference for the optimization problem. It is used for the `mini` and `mini!` enviroments. In the `mini*` environment should be left blank, i.e. {}, **not to be ommited**.

(viii) `optimizationResult`: a term expressing the result of the optimization problem, e.g. $J(w^*) =$. If not needed leave it blank, **not to be ommited**.

The last two defined problem parameters, `\label{optimizationProblem}` and `optimizationResult`, could be made optional. However, in order to improve the problem readibility, line breaking between the 7 parametes was implemented; unfortunately, linea breaking and optional parameters are not compatible and these two parameters had to be made mandatory.

3.2 Adding Constraints

After the definition of the problem parameters, the environment accepts the definition of an infinite number of constraints. For this definitions the following command is used:

    \addConstraint{LHS.k}{RHS.k\label{Const.k}}{extraConst.k}

The command accepts three different parameters

1. `LHS.k`: the left-hand side of the the constraint k, e.g. $3w^\top w$.

2. (Optional) `RHS.k\label{Const.k}`: the right-hand side of the constraint k if the equations should be aligned in the equality or inequality signs, e.g. $\leq \|w\|_\infty$. If required, the constraint label should also be included in this term.

3. (Optional) `extraConst.k`: optional parameter to add extra alignment point for additional constraint information. An example would be the constraint names. Look Example 12.10 or the Section 6.5.

3.2.1 Constraints referencing

Notice that the label for the constraints is always included in the right hand side expression and it only makes sense for the case of using the `mini!` enviroment. The label of the objective function can also be included in a similar way.

4 Environment Types

There are four basic environments depending on the type of referencing that should be used.

1. The **mini** environment for defining problems with a single reference label:

$$
\begin{aligned}
\underset{w}{\text{minimize}} \quad & f(w) + R(w + 6x) \\
& + L(x) \\
\text{subject to} \quad & g(w) = 0
\end{aligned}
\tag{1}
$$

2. The **mini*** environment if the problem does not have to be referenced:

$$\underset{w}{\text{minimize}} \quad f(w) + R(w + 6x)$$
$$\text{subject to} \quad g(w) = 0$$

3. The **mini!** environment if each equation should be referenced:

$$\underset{w}{\text{minimize}} \quad f(w) + R(w + 6x) \tag{2a}$$
$$\text{subject to} \quad g(w) = 0 \tag{2b}$$

4. The **minie** environment: same functionality as the **mini!** environment and it replaces **mini!** when using the `optidef` library with some languages in the babel package. For further details we refer to Section 11.2.

Additionally, there are four basic definitions of optimization problems:

1. The **mini** environment:

$$\underset{w}{\text{minimize}} \quad f(w) + R(w + 6x)$$
$$\text{subject to} \quad g(w) = 0 \tag{3}$$

2. The **maxi** environment:

$$\underset{w}{\text{maximize}} \quad f(w) + R(w + 6x)$$
$$\text{subject to} \quad g(w) = 0 \tag{4}$$

3. The **argmini** environment:

$$\underset{w}{\arg\min} \quad f(w) + R(w + 6x)$$
$$\text{subject to} \quad g(w) = 0 \tag{5}$$

4. The **argmaxi** environment:

$$\underset{w}{\arg\max} \quad f(w) + R(w + 6x)$$
$$\text{subject to} \quad g(w) = 0 \tag{6}$$

5 Long and Short Output Formats

The library permits the definition of two different problem size: a long format and a short format.

5.1 Long Format

Selected by `sizeFormat`=l. It makes use of *subject to* and *minimize/maximize*

$$\underset{w}{\text{minimize}} \quad f(w) + R(w + 6x)$$
$$\text{subject to} \quad g(w) = 0$$

5.2 Short Format

Selected by `sizeFormat`=s. It uses instead the shorter *s.t.* and *min/max*

$$\min_{w} \quad f(w) + R(w + 6x)$$
$$\text{s.t.} \quad g(w) = 0$$

By the default the long format is used. To change the default to the short format the package must be imported with the `short` option:

```
\usepackage[short]{optidef}
```

6 Output Formats for the Constraints

There are four basic output formats for the location of the constraints. They are controlled by the environment parameter `constraintFormat`.

6.1 Alternative 0

In this format option, the constraints are located to the right of *subject to* and aligned with the objective function. It also has a second alignment point at the $=$, $\leq$, $\geq$ signs:

$$\underset{w}{\text{minimize}} \quad f(w) + R(w + 6x)$$
$$\text{subject to} \quad g(w) + h(w) = 0, \tag{7}$$
$$t(w) = 0.$$

It is the default format if no format option is provided. Alternatively, it can also be set by selecting `constraintFormat`=0.

6.2 Alternative 1

Selected by `constraintFormat`=1. It locates the constraints below *subject to* and keeps them aligned at the inequality/equality signs:

$$\underset{w}{\text{minimize}} \quad f(w) + R(w + 6x)$$
$$\text{subject to}$$
$$g(w) + h(w) = 0, \tag{8}$$
$$t(w) = 0.$$

6.3 Alternative 2

Selected by `constraintFormat`=2. It aligns all the constraints with the objective function.

$$\underset{w}{\text{minimize}} \quad f(w) + R(w + 6x)$$
$$\text{subject to} \quad g(w) + h(w) = 0, \tag{9}$$
$$t(w) = 0.$$

6.4 Alternative 3

Selected by `constraintFormat`=3. It aligns all the constraints below *subject to*:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & f(w) + R(w + 6x) \\ \text{subject to} \quad & \\ & g(w) + h(w) = 0, \\ & t(w) = 0. \end{aligned} \tag{10}$$

```
\usepackage[c1|c2|c3]{optidef}
```

6.5 Extra alignment alternative

By default, the constraints have 2 aligned elements. However, a third alignment point can be used to set some constraint features. A clear example could be the constraints names:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & f(w) + R(w + 6x) \\ \text{subject to} \quad & g(w) + h(w) = 0, && \text{(Topological Constraint)}, \\ & l(w) = 5w, && \text{(Boundary Constraint)} \end{aligned}$$

or the index of the constraints:

$$\begin{aligned} \underset{w,\,u}{\text{minimize}} \quad & f(w) + R(w + 6x) \\ \text{subject to} \quad & g(w_k) + h(w_k) = 0, && k = 0, \ldots, N-1, \\ & l(w_k) = 5u, && k = 0, \ldots, N-1 \end{aligned}$$

This extra alignment point can be added using a third input parameter on the `\addConstraint` parameter. An example using the last constraint of the previous example would be:

```
\addConstraint{l(w_k)}{=5u,\quad}{k=0,\ldots,N-1}
```

6.6 Default format

The default format is alternative 0. To change the default format across the whole document, the package can be imported using one of the three options: `c1`, `c2`, `c3`, i.e.:

7 Breaking the optimization problem across multiple pages

In several cases, people encounter the problem of having an optimization problem that is too long to fit in a single page. In those cases, optidef can automatically break the problem across multiple pages by simply using the optional argument <b>. For example:

9

```
\begin{mini*}<b>
{w,u}{f(w)+ R(w+6x)+ H(100w-x*w/500)}{}{}
\addConstraint{g(w_k)+h(w_k)}{=0,}{k=0,\ldots,N-1}
\addConstraint{l(w_k)}{=5u,\quad}{k=0,\ldots,N-1}
\end{mini*}
```

However, when using this option `<b>`, it is important to note that labeling of equations is no longer automatic. To create the number/label, the command `\labelOP{label}` should be used. In particular, in the equation/constraint of the optimization problem where the label/number should be located, simply add `\labelOP{label}`. For example, the following code:

```
\begin{mini}<b>
{w,u}{f(w)+ R(w+6x)+ H(100w-x*w/500)}{}{}
\addConstraint{g(w_k)+h(w_k)}{=0,}{k=0,\ldots,N-1 \
    labelOP{eq:label}}
\addConstraint{l(w_k)}{=5u,\quad}{k=0,\ldots,N-1}
\end{mini}
```

would display this:

$$
\begin{aligned}
\underset{w,\,u}{\text{minimize}} \quad & f(w) + R(w+6x) + H(100w - x*w/500) \\
\text{subject to} \quad & g(w_k) + h(w_k) = 0, \quad k = 0,\ldots,N-1, \\
& l(w_k) = 5u, \quad k = 0,\ldots,N-1
\end{aligned}
$$

(11)

The option `<b>` automatically breaks the optimization problem when the problem is too large to fit in one page (e.g. see an example in 12.13). However, manual breaks at selected locations are also possible using the `\displaybreak` command. Just add `\displaybreak` between the two constraints that need to be broken, e.g.:

```
\begin{mini}<b>
{w,u}{f(w)+ R(w+6x)+ H(100w-x*w/500)}{}{}
\breakObjective{-g(w^3-x^2*200+10000*w^5)}
\addConstraint{g(w_k)+h(w_k)}{=0,}{k=0,\ldots,N-1 \
    labelOP{eq:label}}
\displaybreak
\addConstraint{l(w_k)}{=5u,\quad}{k=0,\ldots,N-1}
\end{mini}
```

would display:

$$
\begin{aligned}
\underset{w,\,u}{\text{minimize}} \quad & f(w) + R(w+6x) + H(100w - x*w/500) \\
& - g(w^3 - x^2 * 200 + 10000 * w^5) \\
\text{subject to} \quad & g(w_k) + h(w_k) = 0, \qquad\qquad k = 0,\ldots,N-1,
\end{aligned}
$$

(12)

$$l(w_k) = 5u, \qquad k = 0, \ldots, N-1$$

8 Breaking the objective across several lines

In several cases, people encounter the problem of having an optimization problem which objective function is too long to be set in a single line. In such cases, a line breaking that respects the rest of the problem syntax would be desirable. To account for that, the command \breakObjective can be used. The idea is that, if the objective function shall be split in n different functions, e.g. $f_1, \ldots, f_n$, the default objective parameter would include just f_1 and then, we would include $n-1$ statements \breakObjective(f_k), $\forall k = 2, \ldots, n$ right before defining the \addConstraint commands.

Let's illustrate this with an example. We could consider the example from before:

$$\begin{aligned} \underset{w,\,u}{\text{minimize}} \quad & f(w) + R(w + 6x) \\ \text{subject to} \quad & g(w_k) + h(w_k) = 0, \quad k = 0, \ldots, N-1, \\ & l(w_k) = 5u, \quad k = 0, \ldots, N-1 \end{aligned} \tag{13}$$

If now the cost function were too long, i.e:

$$f(w) + R(w + 6x) + H(100w - x * w/500) - g(w^3 - x^2 * 200 + 10000 * w^5)$$

We could split it as:

$$\begin{aligned} \underset{w,\,u}{\text{minimize}} \quad & f(w) + R(w + 6x) + H(100w - x * w/500) \\ & - g(w^3 - x^2 * 200 + 10000 * w^5) \\ \text{subject to} \quad & g(w_k) + h(w_k) = 0, \qquad\qquad k = 0, \ldots, N-1, \\ & l(w_k) = 5u, \qquad\qquad k = 0, \ldots, N-1 \end{aligned} \tag{14}$$

by simpling using the following command:

```
\begin{mini*}
{w,u}{f(w)+ R(w+6x)+ H(100w-x*w/500)}{}{}
\breakObjective{-g(w^3-x^2*200+10000*w^5)}
\addConstraint{g(w_k)+h(w_k)}{=0,}{k=0,\ldots,N-1}
\addConstraint{l(w_k)}{=5u,\quad}{k=0,\ldots,N-1}
\end{mini*}
```

It is important to notice the specific location of the \breakObjective command. In order to work properly, it has to be defined right before \addConstraint and right after the definition of the environment parameters; i.e. in any case the command should be used right after defining the first part of the objective function and not finishing the definition of the mandatory environment parameters.

9 Default comma at the end of the constraint

By default, the algorithms adds a comma at the end of any constraint that is not the last one. This feature was implemented due to correctness of mathematical

notation. However, this behavior can be removed by adding the option `nocomma` when importing the package:

```
\usepackage[nocomma]{optidef}
```

10   Long Optimization Variables

The standard appearance for long optimization variables is as follows:

$$\underset{x_0, u_0, x_1, \ldots, u_{N-1}, x_N}{\text{minimize}} \quad \sum_{k=0}^{N-1} L(x_k, u_k) + E(x_N) \tag{15a}$$

$$\text{subject to} \quad x_{k+1} - f(x_k, u_k) = 0, \quad k = 0, \ldots, N-1, \tag{15b}$$

$$h(x_k, u_k) \leq 0, \quad k = 0, \ldots, N-1, \tag{15c}$$

$$r(x_0, x_N) = 0. \tag{15d}$$

A possible way to reduce the large variable spacing is to stack them with the command:

```
\substack{x_0,u_0,x_1,\hdots,\\u_{N-1},x_N}
```

$$\underset{\substack{x_0, u_0, x_1, \ldots, \\ u_{N-1}, x_N}}{\text{minimize}} \quad \sum_{k=0}^{N-1} L(x_k, u_k) + E(x_N) \tag{16a}$$

$$\text{subject to} \quad x_{k+1} - f(x_k, u_k) = 0, \quad k = 0, \ldots, N-1, \tag{16b}$$

$$h(x_k, u_k) \leq 0, \quad k = 0, \ldots, N-1, \tag{16c}$$

$$r(x_0, x_N) = 0. \tag{16d}$$

11   Compatibility issues with other packages

Issues with three different packages have been reported: cleveref, babel, and mathabx.

11.1   Cleveref

When using the cleveref package in couple with the optidef package two measures have to taken for the packages to work properly:

1. As also indicated in the cleveref documentation, the optidef package has to be loaded before the cleveref package.

2. To avoid crashes, the `\label` commands in the optidef environments have to be replaced by the protected counterparts `\protect\label`. This is required because of the standard Latex issue of moving arguments and fragile commands[1].

---

[1] `goo.gl/wmKbNU`

A code example taking into account both measures is the following:

```
\documentclass{article}
\usepackage{optidef}
\usepackage{cleveref}

\begin{document}

\begin{mini!}
  {w}{f(w)+ R(w+6x) \protect\label{eq:ObjectiveExample1}}
  {\label{eq:Example1}}{}
  \addConstraint{g(w)}{=0 \protect\label{eq:C1Example3}}
  \addConstraint{n(w)}{= 6 \protect\label{eq:C2Example1}}
  \addConstraint{L(w)+r(x)}{=Kw+p \protect\label{eq:C3Example1}}
\end{mini!}

Example labels: \cref{eq:Example1} and \cref{eq:ObjectiveExample1}.

\end{document}
```

As an alternative to the second step, i.e. protecting the `\label` command, the command can be robustify in the document preamble and then `\protect` is not longer needed. To robustify the `\label` command, the following has to be added to the preamble:

```
\usepackage{etoolbox}
\robustify{\label}
```

11.2   Babel

When importing the package babel with some specific languages, e.g. French, the `mini!` environment clashes because of the exclamation mark.

This issue has been resolved starting from Optidef 2.7, where a working alternative to the `mini!` environment is included: the `minie` enviroment. Both environemnts have the same functionality, but when using the babel package it is recommended to use the `minie` environment to avoid issues.

11.3   Mathabx

When using the mathabx package in couple with the optidef package, the optidef package must be loaded first in order to avoid malfunction of the mathabx package. In addition, the amsmath package should also be loaded before both of them. The preamble should look like:

```
\usepackage{amsmath}
\usepackage{mathabx}
\usepackage{optidef}
```

12 Examples

12.1 Example 1 - mini environment

The code:

```
\begin{mini}
{w}{f(w)+ R(w+6x)}
{\label{eq:Example1}}{}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{mini}
```

outputs:

$$
\begin{aligned}
\operatorname*{minimize}_{w} \quad & f(w) + R(w + 6x) \\
\text{subject to} \quad & g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned} \tag{17}
$$

12.2 Example 2 - mini* environment

On the other hand:

```
\begin{mini*}
{w}{f(w)+ R(w+6x)}
{}{}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6,}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{mini*}
```

it is almost the same but removing the reference:

$$
\begin{aligned}
\operatorname*{minimize}_{w} \quad & f(w) + R(w + 6x) \\
\text{subject to} \quad & g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned}
$$

12.3   Example 3 - mini! environment

Finally, the multireferencing environment outputs:

```
\begin{mini!}
{w}{f(w)+ R(w+6x) \label{eq:ObjectiveExample1}}
{\label{eq:Example1}}{}

\addConstraint{g(w)}{=0 \label{eq:C1Example3}}
\addConstraint{n(w)}{= 6 \label{eq:C2Example1}}
\addConstraint{L(w)+r(x)}{=Kw+p \label{eq:C3Example1}}
\addConstraint{h(x)}{=0. \label{eq:C4Example1}}
\end{mini!}
```

$$
\begin{aligned}
&\underset{w}{\text{minimize}} && f(w) + R(w + 6x) && \text{(18a)} \\
&\text{subject to} && g(w) = 0, && \text{(18b)} \\
&&& n(w) = 6, && \text{(18c)} \\
&&& L(w) + r(x) = Kw + p, && \text{(18d)} \\
&&& h(x) = 0. && \text{(18e)}
\end{aligned}
$$

12.4   Example 4 - Problem Result

Adding the problem result:

```
\begin{mini}
{w}{f(w)+ R(w+6x)}
{\label{eq:Example1}}
{J(w^*)=}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{mini}
```

outputs:

$$
\begin{aligned}
J(w^*) \;=\; &\underset{w}{\text{minimize}} && f(w) + R(w + 6x) \\
&\text{subject to} && g(w) = 0, \\
&&& n(w) = 6, \\
&&& L(w) + r(x) = Kw + p, \\
&&& h(x) = 0.
\end{aligned}
\qquad (19)
$$

12.5   Example 5 - Short Format

Adding the short format parameter:

```
\begin{mini}|s|
{w}{f(w)+ R(w+6x)}
{\label{eq:Example1}}
{}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{mini}
```

outputs:

$$
\begin{aligned}
\min_{w} \quad & f(w) + R(w + 6x) \\
\text{s.t.} \quad & g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned}
\tag{20}
$$

12.6 Example 6 - Alternative 1 for Constraints

If including a 1 as optional parameter, the first constraint will appear aligned to the left right below *subject to*.

```
\begin{mini}[1]
{w}{f(w)+ R(w+6x)}
{\label{eq:Example1}}
{}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{mini}
```

outputs:

$$
\begin{aligned}
\operatorname*{minimize}_{w} \quad & f(w) + R(w + 6x) \\
\text{subject to} \quad & \\
& g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned}
\tag{21}
$$

12.7 Example 7 - Alternative 2 for Constraints

If including a 2 as optional parameter, the constraint will appear to the right of *subject to* but a single alignment point.

```
\begin{mini}[2]
{w}{f(w)+ R(w+6x)}
{\label{eq:Example1}}
{}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{mini}
```

outputs:

$$
\begin{aligned}
\underset{w}{\text{minimize}} \quad & f(w) + R(w + 6x) \\
\text{subject to} \quad & g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned}
\tag{22}
$$

12.8   Example 8 - Alternative 3 for Constraints

If including a 3 as optional parameter, the first constraint will appear aligned to the left right below *subject to* and with a single alignment point.

```
\begin{mini}[3]
{w}{f(w)+ R(w+6x)}
{\label{eq:Example1}}
{}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{mini}
```

outputs:

$$
\begin{aligned}
\underset{w}{\text{minimize}} \quad & f(w) + R(w + 6x) \\
\text{subject to} \quad & \\
& g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned}
\tag{23}
$$

12.9   Example 9 - Breaking a long objective

```
\begin{mini*}
{w,u}{f(w)+ R(w+6x)+ H(100w-x*w/500)}{}{}
\breakObjective{-g(w^3-x^2*200+10000*w^5)}
\addConstraint{g(w_k)+h(w_k)}{=0,}
\addConstraint{l(w_k)}{=5u,\quad}
\end{mini*}
```

outputs:

$$
\begin{aligned}
\operatorname*{minimize}_{w,\,u} \quad & f(w) + R(w + 6x) + H(100w - x * w/500) \\
& - g(w^3 - x^2 * 200 + 10000 * w^5) \\
\text{subject to} \quad & g(w_k) + h(w_k) = 0, \\
& l(w_k) = 5u.
\end{aligned}
\tag{24}
$$

12.10   Example 10 - Extra Alignment in the Constraints

Adding optional alignment to add constraint names:

```
\begin{mini*}
{w}{f(w)+ R(w+6x)}
{}{}
\addConstraint{g(w)}{=0,}{ \quad  \text{(Dynamic constraint)}}
\addConstraint{n(w)}{= 6,}{ \quad  \text{(Boundary constraint)}}
\addConstraint{L(w)+r(x)}{=Kw+p,}{ \quad  \text{(Random constraint)}}
\addConstraint{h(x)}{=0,}{ \quad  \text{(Path constraint).}}
\end{mini*}
```

12.11   Example 11 - The *argmini* Environment

Similar to the `mini`, `mini*` and `mini!` environments, the environments `argmini`, `argmini*` and `argmini!` are very similar environments that use the same syntax but the output is slightly different:

```
\begin{argmini}
{w}{f(w)+ R(w+6x)}

{\label{eq:Example1}}{w^*=}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{argmini}
```

outputs:

$$
\begin{aligned}
w^* \;=\; & \arg\min_{w} \quad f(w) + R(w + 6x) \\
& \text{subject to} \quad\quad g(w) = 0, \\
& \quad\quad\quad\quad\quad\quad\; n(w) = 6, \\
& \quad\quad\quad\quad\; L(w) + r(x) = Kw + p, \\
& \quad\quad\quad\quad\quad\quad\;\; h(x) = 0.
\end{aligned}
\tag{25}
$$

12.12 Example 12 - The *maxi* and *argmaxi* Environments

Exactly the same syntax and definition as the previous environments, but now for defining maximization environments. The following code serves for illustration:

```
\begin{maxi}
{w}{f(w)+ R(w+6x)}
{g(w)}{=0}

{\label{eq:Example1}}{}

\addConstraint{g(w)}{=0}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{maxi}
```

outputs:

$$
\begin{aligned}
\underset{w}{\text{maximize}} \quad & f(w) + R(w + 6x) \\
\text{subject to} \quad & g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned}
\tag{26}
$$

12.13 Example 13 - Breaking optimization problem

```
\begin{mini}<b>
{w}{f(w)+ R(w+6x)}
{\label{eq:Example1}}{}
\addConstraint{g(w)}{=0}
\addConstraint{p(w)}{=0}
\addConstraint{q(w)}{=0}
\addConstraint{r(w)}{=0\labelOP{testLabel}}
\addConstraint{n(w)}{= 6}
\addConstraint{L(w)+r(x)}{=Kw+p}
\addConstraint{h(x)}{=0.}
\end{mini}
```

outputs:

$$
\begin{aligned}
\underset{w}{\text{minimize}} \quad & f(w) + R(w + 6x) \\
\text{subject to} \quad & g(w) = 0, \\
& p(w) = 0, \\
& q(w) = 0, \\
& r(w) = 0, \\
& n(w) = 6,
\end{aligned}
\tag{27}
$$

$$L(w) + r(x) = Kw + p,$$
$$h(x) = 0.$$

12.14   Example 14 - All Possible Parameters

```
\begin{mini!}|s|[2]<b>
{w}{f(w)+ R(w+6x)\label{eq:ObjectiveExample3}}
{\label{eq:Example3}}
{w^*=}
\addConstraint{g(w)}{=0 \label{eq:C1Example3}}
\addConstraint{n(w)}{= 6 \label{eq:C2Example3}}
\addConstraint{L(w)+r(x)}{=Kw+p \label{eq:C3Example3}}
\addConstraint{h(x)}{=0.\label{eq:C4Example3}}
\end{mini!}
```

$$
\begin{aligned}
w^* = \ \min_{w} \quad & f(w) + R(w + 6x) & \text{(28a)} \\
\text{s.t.} \quad & g(w) = 0, & \text{(28b)} \\
& n(w) = 6, & \text{(28c)} \\
& L(w) + r(x) = Kw + p, & \text{(28d)} \\
& h(x) = 0. & \text{(28e)}
\end{aligned}
$$

13   Reporting bugs and feature requests

To report any bug or request some feature please use the issue section in the github repository: `https://github.com/jeslago/optidef/issues`.