Optidef

A Latex library for optimization problems

Version - 2.6

Jesus Lago

November 29, 2017

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

2 Using the package

The package can be imported by directly adding

```
\usepackage{optidef}
```

to the document preamble. When importing the packages two options can be used, `short` and `nocomma`:

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

For an explanation of the `short` option check Section 5. For the `nocomma` option check Section 8. 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]
{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) `optimizationVariable`: variable to be optimizated in the problem, e.g. $w \in \Re^N$.

(v) `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

(vi) `\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**.

(vii) `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 9.10 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 three basic environments depending on the type of referencing that should be used.

1. The **mini** environment for defining problems with a single reference label:
$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) \\ & + L(x) \\ \text{s.t.} \quad & g(w) = 0 \end{aligned} \tag{1}$$
2. The **mini*** environment if the problem does not have to be referenced:
$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w) = 0 \end{aligned}$$
3. The **mini!** environment if each equation should be referenced:
$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) && \text{(2a)} \\ \text{s.t.} \quad & g(w) = 0 && \text{(2b)} \end{aligned}$$

Additionally, there are four basic definitions of optimization problems:

1. The **mini** environment:

$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w) = 0 \end{aligned} \tag{3}$$

2. The **maxi** environment:

$$\begin{aligned} \max_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w) = 0 \end{aligned} \tag{4}$$

3. The **argmini** environment:

$$\begin{aligned} \arg\min_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w) = 0 \end{aligned} \tag{5}$$

4. The **argmaxi** environment:

$$\begin{aligned} \arg\max_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w) = 0 \end{aligned} \tag{6}$$

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*

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

5.2 Short Format

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

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

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 Standard Format

It is the default format and if `constraintFormat` left blank it is used. Alternatively can be also set by selecting `constraintFormat`=0.

By default the constraints are aligned with the objective function, to the right of *subject to* and with a second alignment point at the $=$, $\leq$, $\geq$:

$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w) + h(w) = 0, \\ & t(w) = 0. \end{aligned} \tag{7}$$

6.2 Alternative 1

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

$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & \\ g(w) + h(w) &= 0, \\ t(w) &= 0. \end{aligned} \tag{8}$$

6.3 Alternative 2

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

$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w) + h(w) = 0, \\ & t(w) = 0. \end{aligned} \tag{9}$$

6.4 Alternative 3

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

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

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:

$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w) + h(w) = 0, \quad \text{(Topological Constraint)}, \\ & l(w) = 5w, \quad \text{(Boundary Constraint)} \end{aligned}$$

or the index of the constraints:

$$\begin{aligned} \min_{w,\,u} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w_k) + h(w_k) = 0, \quad k = 0, \dots, N-1, \\ & l(w_k) = 5u, \quad k = 0, \dots, N-1 \end{aligned}$$

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}
```

7 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. $f_1, \dots, f_n$, the default objective parameter would include just f_1 and then, we would include $n-1$ statements `\breakObjective`(f_k), $\forall k = 2, \dots, n$ right before defining the `\addConstraint` commands.

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

$$\begin{aligned} \min_{w,\,u} \quad & f(w) + R(w + 6x) \\ \text{s.t.} \quad & g(w_k) + h(w_k) = 0, \quad k = 0, \dots, N-1, \\ & l(w_k) = 5u, \quad k = 0, \dots, N-1 \end{aligned} \tag{11}$$

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:

$$\begin{aligned} \min_{w,\,u} \quad & f(w) + R(w + 6x) + H(100w - x * w/500) \\ & - g(w^3 - x^2 * 200 + 10000 * w^5) \\ \text{s.t.} \quad & g(w_k) + h(w_k) = 0, \quad k = 0, \dots, N-1, \\ & l(w_k) = 5u, \quad k = 0, \dots, N-1 \end{aligned} \tag{12}$$

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.

8 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}
```

9 Examples

9.1 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:

$$\begin{aligned} \min_{w} \quad & f(w) + R(w+6x) \\ \text{s.t.} \quad & g(w) = 0, \\ & n(w) = 6, \\ & L(w) + r(x) = Kw + p, \\ & h(x) = 0. \end{aligned} \tag{13}$$

9.2 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:

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

9.3 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!}
```

$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) && \text{(14a)} \\ \text{s.t.} \quad & g(w) = 0, && \text{(14b)} \\ & n(w) = 6, && \text{(14c)} \\ & L(w) + r(x) = Kw + p, && \text{(14d)} \\ & h(x) = 0. && \text{(14e)} \end{aligned}$$

9.4 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:

$$\begin{aligned} J(w^*) \;=\; \min_{w} \quad & f(w) + R(w+6x) \\ \text{s.t.} \quad & g(w) = 0, \\ & n(w) = 6, \\ & L(w) + r(x) = Kw + p, \\ & h(x) = 0. \end{aligned} \tag{15}$$

9.5 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:

$$\begin{aligned} \min_{w} \quad & f(w) + R(w+6x) \\ \text{s.t.} \quad & g(w) = 0, \\ & n(w) = 6, \\ & L(w) + r(x) = Kw + p, \\ & h(x) = 0. \end{aligned} \tag{16}$$

9.6 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:

$$\begin{aligned}
\min_{w} \quad & f(w) + R(w + 6x) \\
\text{s.t.} \quad & \\
& g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned} \tag{17}$$

9.7 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:

$$\begin{aligned}
\min_{w} \quad & f(w) + R(w + 6x) \\
\text{s.t.} \quad & g(w) = 0, \\
& n(w) = 6, \\
& L(w) + r(x) = Kw + p, \\
& h(x) = 0.
\end{aligned} \tag{18}$$

9.8 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:

$$\begin{aligned} \min_{w} \quad & f(w) + R(w + 6x) \\ & \text{s.t.} \\ & g(w) = 0, \\ & n(w) = 6, \\ & L(w) + r(x) = Kw + p, \\ & h(x) = 0. \end{aligned} \tag{19}$$

9.9 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:

$$\begin{aligned} \min_{w,\,u} \quad & f(w) + R(w + 6x) + H(100w - x * w/500) \\ & \quad - g(w^3 - x^2 * 200 + 10000 * w^5) \\ \text{s.t.} \quad & g(w_k) + h(w_k) = 0, \\ & \qquad l(w_k) = 5u. \end{aligned} \tag{20}$$

9.10 Example 9 - 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*}
```

9.11 Example 10 - 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:

$$\begin{aligned} w^* \;=\; \underset{w}{\arg\min} \quad & f(w) + R(w+6x) \\ \text{s.t.} \quad & g(w) = 0, \\ & n(w) = 6, \\ & L(w) + r(x) = Kw + p, \\ & h(x) = 0. \end{aligned} \tag{21}$$

9.12 Example 11 - 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:

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

9.13 Example 12 - All Possible Parameters

```
\begin{mini*}|s|[1]
{w}{f(w)+ R(w+6x)}
{}{w^*=}
\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*}
```

$$
\begin{aligned}
w^* = \min_{w} \quad & f(w) + R(w + 6x) && \text{(23a)}\\
\text{s.t.} \quad & g(w) = 0, && \text{(23b)}\\
& n(w) = 6, && \text{(23c)}\\
& L(w) + r(x) = Kw + p, && \text{(23d)}\\
& h(x) = 0. && \text{(23e)}
\end{aligned}
$$

10 Long Optimization Variables

The standard appearance for long optimization variables is as follows:

$$
\begin{aligned}
\min_{x_0, u_0, x_1, \ldots, u_{N-1}, x_N} \quad & \sum_{k=0}^{N-1} L(x_k, u_k) + E(x_N) && \text{(24a)}\\
\text{s.t.} \quad & x_{k+1} - f(x_k, u_k) = 0, \quad k = 0, \ldots, N-1, && \text{(24b)}\\
& h(x_k, u_k) \leq 0, \quad k = 0, \ldots, N-1, && \text{(24c)}\\
& r(x_0, x_N) = 0. && \text{(24d)}
\end{aligned}
$$

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}
```

$$
\begin{aligned}
\min_{\substack{x_0,u_0,x_1,\ldots,\\ u_{N-1},x_N}} \quad & \sum_{k=0}^{N-1} L(x_k, u_k) + E(x_N) && \text{(25a)}\\
\text{s.t.} \quad & x_{k+1} - f(x_k, u_k) = 0, \quad k = 0, \ldots, N-1, && \text{(25b)}\\
& h(x_k, u_k) \leq 0, \quad k = 0, \ldots, N-1, && \text{(25c)}\\
& r(x_0, x_N) = 0. && \text{(25d)}
\end{aligned}
$$

11 Compatibility issues with other packages

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}
```

12 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`.

13 Code definition

```
% optidef - Version 2.6
%
%Copyright 2017 Jesus Lago
%
%This work may be distributed and/or modified under
    the conditions of the LaTeX Project Public License,
     either version 1.3 of this license or (at your
    option) any later version.
%The latest version of this license is in http://www.
    latex-project.org/lppl.txt and version 1.3 or later
     is part of all distributions of LaTeX version
    2005/12/01 or later.
%
%This work has the LPPL maintenance status 'maintained
    '. The Current Maintainer of this work is J. Lago.
%
%E-mail: J.LagoGarcia@tudelft.nl
%
%This work consists of the file optidef.sty.

\NeedsTeXFormat{LaTeX2e}
```

```
\ProvidesPackage{optidef}[2017/11/29 - version=2.6,
   Package for defining optimization problems]

\RequirePackage{environ}
\RequirePackage{mathtools}
\RequirePackage{xifthen}
\RequirePackage{etoolbox}
\RequirePackage{xparse}
\RequirePackage{calc}

%%%%%%%%%%%%%%%%%%%%%%
% DEFINING PACKAGE OPTIONS
%%%%%%%%%%%%%%%%%%%%%%
% Default
\newcommand{\defaultOCPConstraint}{,}
\newcommand{\defaultProblemFormat}{l}

\DeclareOption{short}{
\renewcommand{\defaultProblemFormat}{s}
}

\DeclareOption{long}{
\renewcommand{\defaultProblemFormat}{l}
}

\DeclareOption{nocomma}{
\renewcommand{\defaultOCPConstraint}{}
}

\ProcessOptions\relax

% This command is required to avoid breakdown of the \
   equal fragile command. In particular, before I had
   \equal{#2}{} tp check if argumetn #2 was empty.
   However, if the argument was a bmatrix object the
   command was breaking. Now this command is robust.
\newcommand{\equalsNothing}[3]{%
\ifthenelse{\equal{\unexpanded{#1}}{}}{#2}{#3}%
}

%%%%%%%%%%%%%%%%%%%%%%
% VARIABLES DEFINITION
%%%%%%%%%%%%%%%%%%%%%%

% Toogle to indicate if during the addConstraint
   command the first constraint should be built
   together with "subject to"
\newtoggle{bodyCon}
\toggletrue{bodyCon}
```

```
% If the previous constraints has 3 elements, we avoid
     setting \span\span at the beginning of the next
    constraint. If there is no previous third element,
    \span\span must be included for correct alignment
\newtoggle{previousThird}
\togglefalse{previousThird}
\newcommand{\spanit}{}

% Variable used to define the subject to word for
    short and long versions
\newcommand{\bodySubjectTo}{Unset Subject to}

% Variable used for defining if the long problem
    format or the short problem format is used
\newcommand{\localProblemFormat}{1}

% Variable to storage which type of of local problem
    is being solved
\newcommand{\localProblemType}{minimize}

% Defining variable to storage problem variable
\newcommand{\localOptimalVariable}{}

\newlength\widthInit

%%%%%%%%%%%%%%%%%%%%%%%%
% OBJECTIVE COMMAND DEFINITION
%%%%%%%%%%%%%%%%%%%%%%%%
\newcommand{\bodyobj}[4]
{
\ifthenelse{\isempty{#4}}
{
&\underset{\displaystyle #1}{\mathrlap{\mathrm{#3}}\
    phantom{\mathrm{subject~to}}} \quad #2\span\span\
    span\span
}
{
#4~ &\underset{\displaystyle #1}{\mathrlap{\mathrm
    {#3}}\phantom{\mathrm{subject~to}}} \quad #2\span\
    span\span\span
}
}


%% LONG VERSION "minimize" instead of   "min"
\newcommand{\bodyobjLong}[4]
{
\ifthenelse{\isempty{#4}}
{
```

```
&\mathmakebox[\widthof{$\underset{\displaystyle #1}{\
    mathrm{subject~to}}$}]{\underset{\displaystyle
    #1}{\mathrm{#3}}} \quad #2\span\span\span\span
}
{
#4~ &\mathmakebox[\widthof{$\underset{\displaystyle
    #1}{\mathrm{subject~to}}$}]{\underset{\displaystyle
     #1}{\mathrm{#3}}} \quad #2\span\span\span\span
}
}

%% SHORT VERSION "min" instead of "minimize"
\newcommand{\bodyobjShort}[4]
{
\ifthenelse{\isempty{#4}}
{
&\underset{\displaystyle #1}{\mathrm{#3}} \quad #2\
    span\span\span\span
}
{
#4 ~ &\underset{\displaystyle #1}{\mathrm{#3}} \quad
    #2\span\span\span\span
}
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DEFINITION DIFFERENT TYPE OF BODY CONSTRAINTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% A BODY CONSTRAINT IS THE INITIAL CONSTRAINT DEFINED
     WITH THE 'SUBJECT TO', DEPENDING ON THE TYPE OF
    PROBLEM A DIFFERENT VERSION IS USED


% Main command. Dynamically redefined at every new
    problem definition.
\DeclareDocumentCommand{\bodyconst}{m G{}}
{       \equalsNothing{#2}{
\\ &\underset{\displaystyle \phantom{\
    localOptimalVariable}}{\mathrm{subject~to}} \quad
    &&#1 #2
}{
\\ &\underset{\displaystyle \phantom{\
    localOptimalVariable}}{\mathrm{subject~to}}  \quad
    &#1 & #2
}
}

\newcommand{\bodySubjectToDefinition}{
```

```
%## If the short version of "subject to", i.e. "s.t.",
     should be used the command \bodySubjectTo should
    be modified
\ifthenelse{\equal{\localProblemFormat}{s}}
{%%
\global\def\bodySubjectTo{\mathmakebox[\widthof{$\
    underset{\displaystyle \phantom{\
    localOptimalVariable}}{\mathrm{\localProblemType}}$
    }][c]{\mathmakebox[\widthof{$\mathrm{\
    localProblemType}$}][l]{\mathrm{\kern 0.1em s.t
    .}}}}
}{%%
\global\def\bodySubjectTo{\mathmakebox[\widthof{$\
    underset{\displaystyle \phantom{\
    localOptimalVariable}}{\mathrm{\,subject~to}}$}][c
    ]{\mathmakebox[\widthof{$\mathrm{\localProblemType}
    $}][l]{\mathrm{subject~to}}}}
}%%
%
}

% Standard version.
\DeclareDocumentCommand{\bodyconstRight}{m G{} G{}}
{%%%
\bodySubjectToDefinition
%##  Set the first constraint according to the format
    used for "subject to"
\equalsNothing{#3}{%%
\equalsNothing{#2}{%
\\ & \bodySubjectTo \quad &&#1 #2
}{%
\\ &\bodySubjectTo \quad &#1 & #2
}%
\togglefalse{previousThird}
}{%%
\equalsNothing{#2}{%
\\ &\bodySubjectTo \quad &&#1 #2 &&#3
}{%
\\ &\bodySubjectTo \quad &#1 & #2 &&#3
}%
\toggletrue{previousThird}
}%%
}%%%


% Single alignment point but next to subject to
\DeclareDocumentCommand{\bodyconstOneAlign}{m G{} G{}}
{
\bodySubjectToDefinition
```

```
%##  Set the first constraint according to the format
   used for "subject to"
\equalsNothing{#3}{
\\ &\bodySubjectTo\quad &&#1 #2         \togglefalse{
   previousThird}
}{
\\ &\bodySubjectTo\quad &&#1 #2 &&#3
\toggletrue{previousThird}
}
}

% Contraints below subject to and with a single
   alignment point
\DeclareDocumentCommand{\bodyconstOneAlignBelow}{m G{}
    G{}}
{
\bodySubjectToDefinition
%##  Set the first constraint according to the format
   used for "subject to"
\equalsNothing{#3}{
\\ &\bodySubjectTo \span\span\span\span \\
&&&#1 #2 \togglefalse{previousThird}
}{
\\ &\bodySubjectTo \span\span\span\span \\
&&&#1 #2 &&#3
\toggletrue{previousThird}
}
}

% Contraints below subject to but with double
   alignment point
\DeclareDocumentCommand{\bodyconstBelow}{m G{} G{}}
{
\bodySubjectToDefinition
%##  Set the first constraint according to the format
   used for "subject to"
\equalsNothing{#3}{
\equalsNothing{#2}{
\\ &\bodySubjectTo\span\span\span\span \\
&&&#1 #2
}{
\\ &\bodySubjectTo \span\span\span\span \\
&&#1 & #2
}
\togglefalse{previousThird}
}{
\equalsNothing{#2}{
\\ &\bodySubjectTo \span\span\span\span \\
&&&#1 #2 &&#3
}{
```

```
\\ &\bodySubjectTo\span\span\span\span \\
&&#1 & #2 &&#3
}
\toggletrue{previousThird}
}
}

% Contraints below subject to for the case of having a
    reference/label for each individual equation
\DeclareDocumentCommand{\bodyconstBelowMult}{m G{} G
   {}}
{
\bodySubjectToDefinition
%##  Set the first constraint according to the format
   used for "subject to"
\equalsNothing{#3}{
\equalsNothing{#2}{
\\ &\bodySubjectTo\span\span\span\span \nonumber \\
&&&#1 #2
}{
\\ &\bodySubjectTo \span\span\span\span \nonumber \\
&&#1 & #2
}
\togglefalse{previousThird}
}{
\equalsNothing{#2}{
\\ &\bodySubjectTo\span\span\span\span \nonumber \\
&&&#1 #2        &&#3
}{
\\ &\bodySubjectTo \span\span\span\span \nonumber \\
&&#1 & #2 &&#3
}
\toggletrue{previousThird}
}
}

% Contraints below subject to and with a single
   alignment point for the case of having a reference/
   label for each individual equation
\DeclareDocumentCommand{\bodyconstOneAlignBelowMult}{m
    G{} G{}}
{
\bodySubjectToDefinition
%##  Set the first constraint according to the format
   used for "subject to"
\equalsNothing{#3}{
\\ &\bodySubjectTo\span\span\span\span \nonumber \\
&&&#1 #2                \togglefalse{previousThird}
}{
\\ &\bodySubjectTo\span\span\span\span \nonumber \\
```

```
&&&#1 #2 &&     #3
\toggletrue{previousThird}
}
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DEFINITION DIFFERENT TYPE OF ADDING CONSTRAINTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Main command. Dynamically redefined at every problem
    definiton.
\DeclareDocumentCommand{\addConstraint}{m G{} G{}}{
% "If clause" selecting whether a third parameter (#3)
    defining extra constraint information is used
\equalsNothing{#3}{
% Second "If clause" selecting whether two or 1
   elements for the constraints are used
\equalsNothing{#2}{
\iftoggle{bodyCon}{
\bodyconst{#1}
\togglefalse{bodyCon}
}{
\defaultOCPConstraint\\&\quad &&#1 #2\span\span
\togglefalse{bodyCon}
}
}{
\iftoggle{bodyCon}{
\bodyconst{#1}{#2}
\togglefalse{bodyCon}
}{
\defaultOCPConstraint\\&\quad &#1 & #2\span\span
\togglefalse{bodyCon}
}
}
\togglefalse{previousThird}
}{
\iftoggle{bodyCon}{
\bodyconst{#1}{#2}{#3}
\togglefalse{bodyCon}
}{
\equalsNothing{#2}{
\defaultOCPConstraint\\&\quad &&#1 #2 && #3
}{
\defaultOCPConstraint\\&\quad &#1 & #2 && #3
}
\togglefalse{bodyCon}
}
\toggletrue{previousThird}
```

```
}
}

% Standard version of adding constraints
% The toggle previousThird indicates if the previous
    constraint had three arguments or two. According to
     that it adjust the required spans at the end of
    the previous contraint. It is needed because the
    line jump after a constraint it is done at the
    beginning of the next constraint, not after the
    constraint itself. That avoid the last jump of the
    last constraint.
% The toggle bodyCon indicate that it is the first
    contraint. The first contraint is different that
    the rest because it contains "subject to"
% The commands \equalsNothing checks if the optional
    arguments exists
\DeclareDocumentCommand{\standardAddConstraint}{m G{}
    G{}}{
\iftoggle{previousThird}
{
\renewcommand{\spanit}{}
}{
\renewcommand{\spanit}{\span\span}
}
\iftoggle{bodyCon}{
\bodyconstRight{#1}{#2}{#3}
\togglefalse{bodyCon}
}{
\equalsNothing{#2}{
\equalsNothing{#3}{
\defaultOCPConstraint\spanit\\&\quad &&#1 #2
\togglefalse{previousThird}
}{
\defaultOCPConstraint\spanit\\&\quad &&#1 #2 && #3
\toggletrue{previousThird}
}
}{
\equalsNothing{#3}{
\defaultOCPConstraint\spanit\\&\quad &#1 & #2
\togglefalse{previousThird}
}{
\defaultOCPConstraint\spanit\\&\quad &#1 & #2 && #3
\toggletrue{previousThird}
}
}
\togglefalse{bodyCon}
}
}
```

```
% Adding constraints below subject to
\DeclareDocumentCommand{\BelowAddConstraint}{m G{} G
   {}}{
\iftoggle{bodyCon}{
\bodyconstBelow{#1}{#2}{#3}
\togglefalse{bodyCon}
}{
\equalsNothing{#2}{
\equalsNothing{#3}{
\defaultOCPConstraint\spanit\\&&&#1  #2
          \togglefalse{previousThird}
}{
\defaultOCPConstraint\spanit\\&&&#1  #2 && #3
\toggletrue{previousThird}
}
}{
\equalsNothing{#3}{
\defaultOCPConstraint\spanit\\ &&#1  &#2 \togglefalse{
   previousThird}
}{
\defaultOCPConstraint\spanit\\ &&#1  &#2 && #3
\toggletrue{previousThird}
}
}
\togglefalse{bodyCon}
}
}

% Adding constraints with a single alignment point but
    next to subject to
\DeclareDocumentCommand{\oneAlignAddConstraint}{m G{}
   G{}}{
\iftoggle{bodyCon}{
\bodyconstOneAlign{#1}{#2}{#3}
\togglefalse{bodyCon}
}{
\equalsNothing{#3}{
\defaultOCPConstraint\spanit\\&\quad &&#1  #2 \
   togglefalse{previousThird}
}{
\defaultOCPConstraint\spanit\\&\quad &&#1  #2 && #3
\toggletrue{previousThird}
}
\togglefalse{bodyCon}
}
}

% Adding constraints for a single alignment point and
   with the constraints below
```

```
\DeclareDocumentCommand{\oneAlignBelowAddConstraint}{m
    G{} G{}}{
\iftoggle{bodyCon}{
\bodyconstOneAlignBelow{#1}{#2}{#3}
\togglefalse{bodyCon}
}{
\equalsNothing{#3}{
\defaultOCPConstraint\spanit\\& &&#1  #2\togglefalse{
   previousThird}
}{
\defaultOCPConstraint\spanit\\& &&#1  #2 && #3
\toggletrue{previousThird}
}
\togglefalse{bodyCon}
}
}

% Adding constraints below "subject to" for multiple
   references
\DeclareDocumentCommand{\BelowAddConstraintMult}{m G{}
    G{}}{
\iftoggle{bodyCon}{
\bodyconstBelowMult{#1}{#2}{#3}
\togglefalse{bodyCon}
}{
\equalsNothing{#3}{
\equalsNothing{#2}{
\defaultOCPConstraint\spanit\\&&&#1  #2
}{
\defaultOCPConstraint\spanit\\ &&#1  &#2
}
\togglefalse{previousThird}
}{
\equalsNothing{#2}{
\defaultOCPConstraint\spanit\\&&&#1  #2 && #3
}{
\defaultOCPConstraint\spanit\\ &&#1  &#2&& #3}
\toggletrue{previousThird}
}
\togglefalse{bodyCon}
}
}

% Adding constraints for a single alignment point and
   with the constraints below for multiple references
\DeclareDocumentCommand{\
   oneAlignBelowAddConstraintMult}{m G{} G{}}{
\iftoggle{bodyCon}{
\bodyconstOneAlignBelowMult{#1}{#2}{#3}
\togglefalse{bodyCon}
```

```
}{
\equalsNothing{#3}{
\defaultOCPConstraint\spanit\\& &&#1  #2 \togglefalse{
   previousThird}
}{
\defaultOCPConstraint\spanit\\& &&#1  #2 && #3
\toggletrue{previousThird}
}
\togglefalse{bodyCon}
}
}
%%%%%%%%%%%%%%%%%%%%
% ADDING EXTRA LINE
%%%%%%%%%%%%%%%%%%%%
\newcommand{\breakObjective}[1]
{
\\&\mathmakebox[\widthInit]{\phantom{\underset
   {}{}}}#1\span\span\span\span
}


%%%%%%%%%%%%%%%%%%%%
% SELECTING TYPE OF FORMAT
%%%%%%%%%%%%%%%%%%%%
\newcommand{\selectConstraint}[1]{
\ifthenelse{\equal{#1}{1}}{
\let\addConstraint\BelowAddConstraint
}{
\ifthenelse{\equal{#1}{2}}{
\let\addConstraint\oneAlignAddConstraint
}{
\ifthenelse{\equal{#1}{3}}{
\let\addConstraint\oneAlignBelowAddConstraint
}{
\let\addConstraint\standardAddConstraint}
}
}
}

% Selecting for multiple references
\newcommand{\selectConstraintMult}[1]{
\ifthenelse{\equal{#1}{1}}{
\let\addConstraint\BelowAddConstraintMult
}{
\ifthenelse{\equal{#1}{2}}{
\let\addConstraint\oneAlignAddConstraint
}{
\ifthenelse{\equal{#1}{3}}{
\let\addConstraint\oneAlignBelowAddConstraintMult
}{
```

```
\let\addConstraint\standardAddConstraint}
}
}
}

%%%%%%%%%%%%%%%%%%
% SETTING DEFAULT FORMAT
%%%%%%%%%%%%%%%%%%
% Originally, \toggletrue{bodyCon} was inside this
   function, however, spacing issues after environment
    made me remove it.
\newcommand{\setStandardMini}{
\let\addConstraint\standardAddConstraint
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% COMMANDS TO DEFINE ALL REQUIRED PROPERTIES TO CHOOSE
    SHORT/LONG FORMAT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\newcommand{\setFormatShort}[2]{\global\def\
   localProblemFormat{s} \let\bodyobj\bodyobjShort \
   renewcommand{\localProblemType}{#1}
\setlength{\widthInit}{\widthof{$\underset{\
   displaystyle #2}{\mathrm{#1}}$\quad}}
}

\newcommand{\setFormatLong}[2]{\global\def\
   localProblemFormat{l} \let\bodyobj\bodyobjLong \
   renewcommand{\localProblemType}{#1}
\setlength{\widthInit}{\widthof{$\underset{\
   displaystyle #2}{\mathrm{subject~to}}$ \quad}}
}

%%%%%%%%%%%%%%%%%%%%
%MINIMIZATION ENVIRONMENTS
%%%%%%%%%%%%%%%%%%%%


% BASE ENVIRONMENTS
% Base environment for the three possible types of
   referencing: 1 label, no label or multilabel
% Base environment defined using NewEnviron package
   because of \BODY command
\NewEnviron{BaseMini}[6]{%
\selectConstraint{#1}
\renewcommand{\localOptimalVariable}{#2}
\begin{equation}
#4
```

```
\begin{alignedat}{5}
\bodyobj{#2}{#3}{#6}{#5}
\BODY
\end{alignedat}
\end{equation}
\setStandardMini
}

\NewEnviron{BaseMiniStar}[5]{%
\selectConstraint{#1}
\renewcommand{\localOptimalVariable}{#2}
\begin{alignat*}{5}
\bodyobj{#2}{#3}{#5}{#4}
\BODY
\end{alignat*}
\setStandardMini
}


\NewEnviron{BaseMiniExclam}[6]{%
\selectConstraintMult{#1}
\renewcommand{\localOptimalVariable}{#2}
\begin{subequations}
#4
\begin{alignat}{5}
\bodyobj{#2}{#3}{#6}{#5}
\BODY
\end{alignat}
\end{subequations}
\setStandardMini
}


% INDIVIDUAL AND SPECIFIC ENVIRONMENTS (mini, maxi,
   argmini*...)
% Specific environments defined with xparse package
   due to arguments options

%MINIMIZATION ENVIRONMENTS
% In the below definitions, \toggletrue{bodyCon} has
   to be added once the definition of the environment
   is finished. I tried to do inside the environment
   itself using \setStandardMini, but it produced some
    ugly text displacemente.

% Single reference probems
\DeclareDocumentEnvironment{mini}{D||{\
   defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
```

```
{\setFormatShort{min}{#2} \BaseMini
    {#2}{#3}{#4}{#5}{#6}{min}}
% Long version problem
{\setFormatLong{minimize}{#2} \BaseMini
    {#2}{#3}{#4}{#5}{#6}{minimize}}
}{\endBaseMini\toggletrue{bodyCon}}

\DeclareDocumentEnvironment{argmini}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{arg~min}{#2} \BaseMini
    {#2}{#3}{#4}{#5}{#6}{arg~min}}
% Long version problem
{\setFormatLong{arg~min}{#2} \BaseMini
    {#2}{#3}{#4}{#5}{#6}{arg~min}}
}{\endBaseMini\toggletrue{bodyCon}}


% No reference
\DeclareDocumentEnvironment{mini*}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{min}{#2} \BaseMiniStar
    {#2}{#3}{#4}{#6}{min}}
% Long version problem
{\setFormatLong{minimize}{#2} \BaseMiniStar
    {#2}{#3}{#4}{#6}{minimize}}
}{\endBaseMiniStar\toggletrue{bodyCon}}

\DeclareDocumentEnvironment{argmini*}{D||{l} O{0} m m
    m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{arg~min}{#2}\BaseMiniStar
    {#2}{#3}{#4}{#6}{arg~min}}
% Long version problem
{\setFormatLong{arg~min}{#2} \BaseMiniStar
    {#2}{#3}{#4}{#6}{arg~min}}
}{\endBaseMiniStar\toggletrue{bodyCon}}


% Multiple reference
\DeclareDocumentEnvironment{mini!}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{min}{#2} \BaseMiniExclam
    {#2}{#3}{#4}{#5}{#6}{min}}
```

```
% Long version problem
{\setFormatLong{minimize}{#2} \BaseMiniExclam
    {#2}{#3}{#4}{#5}{#6}{minimize}}
}{\endBaseMiniExclam\toggletrue{bodyCon}}

\DeclareDocumentEnvironment{argmini!}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{arg~min}{#2}\BaseMiniExclam
    {#2}{#3}{#4}{#5}{#6}{arg~min}}
% Long version problem
{\setFormatLong{arg~min}{#2} \BaseMiniExclam
    {#2}{#3}{#4}{#5}{#6}{arg~min}}
}{\endBaseMiniExclam\toggletrue{bodyCon}}



%MAXIMIZATION ENVIRONMENTS

% Single reference probems
\DeclareDocumentEnvironment{maxi}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{max}{#2} \BaseMini
    {#2}{#3}{#4}{#5}{#6}{max}}
% Long version problem
{\setFormatLong{maximize}{#2} \BaseMini
    {#2}{#3}{#4}{#5}{#6}{maximize}}
}{\endBaseMini\toggletrue{bodyCon}}

\DeclareDocumentEnvironment{argmaxi}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{arg~max}{#2} \BaseMini
    {#2}{#3}{#4}{#5}{#6}{arg~max}}
% Long version problem
{\setFormatLong{arg~max}{#2} \BaseMini
    {#2}{#3}{#4}{#5}{#6}{arg~max}}
}{\endBaseMini\toggletrue{bodyCon}}


% No reference
\DeclareDocumentEnvironment{maxi*}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
```

```
{\setFormatShort{max}{#2} \BaseMiniStar
    {#2}{#3}{#4}{#6}{max}}
% Long version problem
{\setFormatLong{maximize}{#2} \BaseMiniStar
    {#2}{#3}{#4}{#6}{maximize}}
}{\endBaseMiniStar\toggletrue{bodyCon}}

\DeclareDocumentEnvironment{argmaxi*}{D||{l} O{0} m m
    m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{arg~max}{#2}\BaseMiniStar
    {#2}{#3}{#4}{#6}{arg~max}}
% Long version problem
{\setFormatLong{arg~max}{#2} \BaseMiniStar
    {#2}{#3}{#4}{#6}{arg~max}}
}{\endBaseMiniStar\toggletrue{bodyCon}}


% Multiple reference
\DeclareDocumentEnvironment{maxi!}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{max}{#2} \BaseMiniExclam
    {#2}{#3}{#4}{#5}{#6}{max}}
% Long version problem
{\setFormatLong{maximize}{#2} \BaseMiniExclam
    {#2}{#3}{#4}{#5}{#6}{maximize}}
}{\endBaseMiniExclam\toggletrue{bodyCon}}

\DeclareDocumentEnvironment{argmaxi!}{D||{\
    defaultProblemFormat} O{0} m m m m}
{\ifthenelse{\equal{#1}{s}}
% Short version problem
{\setFormatShort{arg~max}{#2}\BaseMiniExclam
    {#2}{#3}{#4}{#5}{#6}{arg~max}}
% Long version problem
{\setFormatLong{arg~max}{#2} \BaseMiniExclam
    {#2}{#3}{#4}{#5}{#6}{arg~max}}
}{\endBaseMiniExclam\toggletrue{bodyCon}}
```