

The `mpformulation` package

Typesetting optimization models in L^AT_EX

Version 2.0 — September 28, 2026

© 2025–2026 FYP

Report bugs and comments to mpformulation.sty@gmail.com

Abstract

The `mpformulation` package typesets mathematical programs the way they appear in operations research journals: an objective function, a “subject to” line, and one constraint per line with its quantifier ($\forall i \in I$) flushed right and an equation number in the margin. Four commands and one environment cover the common cases. When a constraint is too wide, for instance in a two-column journal, the package shortens the indent, breaks the quantifier at its commas, or moves it to the next line, without any change to the source.

Contents

1	Introduction	2
2	Getting started	3
3	Writing constraints	4
3.1	Expression and quantifier	4
3.2	Numbering and cross-references	4
3.3	Punctuation	5
3.4	Numbering the whole model	5
4	Objective and “subject to”	6
4.1	Long objectives	7
4.2	“subject to” on the first constraint line	8
5	Adjusting the layout	8
5.1	Setting options	8

5.2	Horizontal and vertical spacing	9
5.3	Aligning on the relation	9
5.4	Left-aligned quantifiers	10
6	Long constraints and narrow columns	11
6.1	Shrinking the indent	12
6.2	Breaking the quantifier automatically	12
6.3	Choosing the break yourself	12
6.4	Moving the quantifier below	13
6.5	Breaking the expression	13
6.6	Fine-tuning	14
7	Models across pages	14
8	A complete example	14
9	Reference	16
9.1	Commands and environment	16
9.2	Options	17
9.3	Macros and lengths	17
9.4	Former option names	18
10	Limitations	18
11	Change history	19

1 Introduction

Optimization models follow a well-established pattern: a model name, an objective, the words “subject to”, then a list of constraints. Each constraint has two parts: an *expression*, the relation itself (such as $x_{ij} \leq y_i$), and a *quantifier*, the set of indices over which it holds (such as $\forall i \in I, j \in J$). Readers expect the relations to start on a common left edge, the quantifiers to be aligned on the right, and every constraint to carry a number they can refer to.

The usual L^AT_EX tools do not produce this layout directly. With `align`, the quantifiers must be pushed right by hand (`\qqquad\forall i`), they are not aligned with each other, and a long constraint in a narrow column overflows or needs manual rework. Tables give alignment but lose equation numbering and cross-references.

`mpformulation` separates the two parts of a constraint, so that the package, not the author, decides

where they go:

- the expression starts at a fixed indent, the quantifier is flushed right, and the number is placed after the quantifier;
- the model name and the objective are aligned with the constraints;
- equation numbers use the standard `equation` counter, so `\label`, `\eqref`, `\cref` work as usual;
- constraints too wide for the line are reorganized automatically ([Section 6](#)).

The first sections teach the package by example. The complete list of commands and options is in [Section 9](#).

2 Getting started

Load the package in the preamble:

```
\usepackage{mpformulation}
```

It loads `amsmath` and `keyval`. `mathtools` (for `\mathclap`) and `cleveref` are optional and used in some examples.

A model is written inside a `mpformulation` environment. Here is the binary knapsack problem:

```
\begin{mpformulation}
  \objective[MODEL1:objective]{KP}{Maximize}{\sum_{j \in J} p_j x_j}
  \constraint[MODEL1:constr1]{\sum_{j \in J} w_j x_j \leq W}{ }
  \constraint[MODEL1:constr2]{x_j \in \{0, 1\}}{\forall j \in J}
\end{mpformulation}
Constraint~\eqref{MODEL1:constr1} bounds the total weight.
```

$$(KP) \quad \text{Maximize} \quad \sum_{j \in J} p_j x_j \tag{1}$$

subject to

$$\sum_{j \in J} w_j x_j \leq W \tag{2}$$

$$x_j \in \{0, 1\} \quad \forall j \in J. \tag{3}$$

Constraint (2) bounds the total weight.

The pattern is always the same:

- `\objective[label]{name}{sense}{expression}` writes the objective line; “subject to” is added automatically before the first constraint;

- `\constraint[⟨label⟩]{⟨expression⟩}{⟨quantifier⟩}` writes one constraint; the quantifier may be empty. The last constraint before `\end{mpformulation}` is detected and ends with a period.

The optional `⟨label⟩` both numbers the line and labels it.

3 Writing constraints

3.1 Expression and quantifier

The first mandatory argument of `\constraint` is the expression, typeset in display style (sums have their limits below). The second is the quantifier, typeset in text style and flushed right. Both are in math mode: do not add `$`.

Separate the index sets of a quantifier with `,\,` (a comma and a thin space). The comma is more than punctuation: it is where the package may break a long quantifier ([Section 6](#)).

```
\begin{mpformulation}
  \constraint[MODEL2:constr1]{\sum_{i \in I} x_{ij} = 1}{\forall j \in J}
  \constraint[MODEL2:constr2]{x_{ij} \leq y_i}{\forall i \in I,\, j \in J}
  \constraint[MODEL2:constr3]{y_i \in \{0, 1\}}{\forall i \in I}
\end{mpformulation}
```

$$\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J \qquad (4)$$

$$x_{ij} \leq y_i \qquad \forall i \in I, j \in J \qquad (5)$$

$$y_i \in \{0, 1\} \qquad \forall i \in I. \qquad (6)$$

3.2 Numbering and cross-references

A constraint is numbered only when it has a label. Without the optional argument (or with an empty one), the line is not numbered and the equation counter is not incremented. Numbers follow the document numbering, including `\numberwithin`.

```
\begin{mpformulation}
  \constraint[MODEL3:constr1]{x_{ij} \leq y_i}{\forall i \in I,\, j \in J}
  \constraint{x_{ij} \geq 0}{\forall i \in I,\, j \in J}
  \constraint[MODEL3:constr2]{y_i \in \{0, 1\}}{\forall i \in I}
\end{mpformulation}
Nonnegativity is not numbered; \cref{MODEL3:constr2} defines the binary
variables and \eqref{MODEL3:constr1} links them to $x$.
```

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J \quad (7)$$

$$x_{ij} \geq 0 \quad \forall i \in I, j \in J$$

$$y_i \in \{0, 1\} \quad \forall i \in I. \quad (8)$$

Nonnegativity is not numbered; Equation (8) defines the binary variables and (7) links them to x .

3.3 Punctuation

Many journals punctuate models as sentences: a comma after each relation and a period at the end. Three options control this: `expr_suffix` is appended to every expression, `quant_suffix` to every quantifier except the last, and `last_suffix` (default “.”) to the quantifier of the last constraint.

The last constraint is the one directly followed by `\end{mpformulation}` (spaces, comments and blank lines in between are ignored). To mark it yourself, for instance when a model is split over two environments or written outside the environment, use `\lastconstraint` instead of `\constraint`; `last_auto=false` turns the detection off.

```
\begin{mpformulation}[expr_suffix={\;,}, quant_suffix={;}]
  \constraint[MODEL4:constr1]{\sum_{i \in I} x_{ij} = 1}{\forall j \in J}
  \constraint[MODEL4:constr2]{x_{ij} \leq y_i}{\forall i \in I, \, j \in J}
\end{mpformulation}
```

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J; \quad (9)$$

$$x_{ij} \leq y_i, \quad \forall i \in I, j \in J. \quad (10)$$

The suffix options are global: set them once in the preamble with `\mpsetup` (Section 5) so that all models of a document agree.

When a constraint has no quantifier, its suffix is attached to the expression, so a last constraint without quantifier ends as “ $x \geq 1$.” and not with a period alone in the margin.

3.4 Numbering the whole model

By default each labelled line has its own number. Two other schemes are available through the `numbering` option of the environment. `numbering=sub` gives the model one number and numbers its lines (1a), (1b), ..., as the `subequations` environment of `amsmath` does. The environment option `label` labels the model itself, and `name_label` labels its name, so that `\ref` prints “(UFLP)”.

```
\begin{mpformulation}[numbering=sub, label=MODEL14, name_label=MODEL14:name]
  \objective[MODEL14:objective]{UFLP}{min}{\sum_{i \in I} f_i y_i + \sum_{i \in I}
    \sum_{j \in J} c_{ij} x_{ij}}
```

```

\constraint[MODEL14:constr1]{\sum_{i \in I} x_{ij} = 1}{\forall j \in J}
\constraint[MODEL14:constr2]{x_{ij} \leq y_i}{\forall i \in I, \forall j \in J}
\end{mpformulation}
Model~\ref{MODEL14:name} is \eqref{MODEL14}; its objective is
\eqref{MODEL14:objective}.

```

$$(UFLP) \quad \min \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (11a)$$

subject to

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J \quad (11b)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J. \quad (11c)$$

Model (UFLP) is (11); its objective is (11a).

`numbering=model` prints a single number for the whole model, on its first line. Line labels, if any, refer to that number.

```

\begin{mpformulation}[numbering=model, label=MODEL15]
\objective{KP}{max}{\sum_{j \in J} p_j x_j}
\constraint{\sum_{j \in J} w_j x_j \leq W}{}
\constraint{x_j \in \{0, 1\}}{\forall j \in J}
\end{mpformulation}
The knapsack problem~\eqref{MODEL15} has a single constraint.

```

$$(KP) \quad \max \quad \sum_{j \in J} p_j x_j \quad (12)$$

subject to

$$\sum_{j \in J} w_j x_j \leq W$$

$$x_j \in \{0, 1\} \quad \forall j \in J.$$

The knapsack problem (12) has a single constraint.

4 Objective and “subject to”

`\objective[⟨label⟩]{⟨name⟩}{⟨sense⟩}{⟨expression⟩}` starts at the same indent as the constraints, so the model name is aligned with the relations below. The name is printed in parentheses; leave it empty to omit it. The sense is ordinary text (*Minimize*, *max*, ...).

```

\begin{mpformulation}

```

```

\objective[MODEL5:objective]{\min}{\sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}}
\constraint[MODEL5:constr1]{\sum_{i \in I} x_{ij} = 1}{\forall j \in J}
\end{mpformulation}

```

$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (13)$$

subject to

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J. \quad (14)$$

The “subject to” line is inserted automatically before the first constraint that follows `\objective`; there is nothing to write. To place it yourself (for instance after a comment line), write `\subjectto`: the automatic line is then skipped. `st_auto=false` turns the automatic line off, for models written without “subject to”.

By default the line is indented like the first line of a paragraph, so it lines up with the text around the model. The key `st_indent` moves it (`Opt` is flush left; `\constraintLeftmargin` aligns it with the constraints), and `st_text` changes the wording. The model name format is the macro `\constraintNameFormat`.

```

\renewcommand{\constraintNameFormat}[1]{\textsc{#1}:}
\begin{mpformulation}[st_text={s.t.}, st_indent=\constraintLeftmargin]
\objective[MODEL6:objective]{UFLP}{\min}{\sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}}
\constraint[MODEL6:constr1]{x_{ij} \leq y_i}{\forall i \in I, j \in J}
\end{mpformulation}

```

$$\text{UFLP: } \min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (15)$$

s.t.

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J. \quad (16)$$

The spaces after the name and after the sense are `name_sep` and `sense_sep` (both `1em`).

4.1 Long objectives

An objective too wide for the line is broken where you put `\\` in the expression. The continuation lines start under the expression and the number goes on the last line. As in constraints, the break is optional: it is used only when the objective does not fit (`obj_break=always` forces it). Start the continuation with `{}`+ to keep the spacing of the operator.

```

\begin{mpformulation}[indent=1em, numsep
=0.5em]
\objective[MODEL16:objective]{TT}{min}{
\sum_{i\in I}\sum_{k\in K} e_{ik}x_{ik}
} + \mu \sum_{t\in T} z_t \\\
\lambda \sum_{i\in I} L_i
\constraint[MODEL16:constr1]{\sum_{k\in
K} x_{ik} = 1}{\forall i \in I}
\end{mpformulation}

```

$$\begin{aligned}
(\text{TT}) \quad \min \quad & \sum_{i \in I} \sum_{k \in K} e_{ik} x_{ik} + \mu \sum_{t \in T} z_t \\
& + \lambda \sum_{i \in I} L_i \quad (17)
\end{aligned}$$

subject to

$$\sum_{k \in K} x_{ik} = 1 \quad \forall i \in I. \quad (18)$$

4.2 “subject to” on the first constraint line

With `st_inline`, “subject to” is written on the line of the first constraint, and the constraints (and the objective) are moved right if needed so that it fits.

```

\begin{mpformulation}[st_inline]
\objective[MODEL17:objective]{UFLP}{min}{\sum_{i \in I} f_i y_i + \sum_{i \in I}
\sum_{j \in J} c_{ij} x_{ij}}
\constraint[MODEL17:constr1]{\sum_{i \in I} x_{ij} = 1}{\forall j \in J}
\constraint[MODEL17:constr2]{x_{ij} \leq y_i}{\forall i \in I, j \in J}
\end{mpformulation}

```

$$(\text{UFLP}) \quad \min \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (19)$$

$$\text{subject to} \quad \sum_{i \in I} x_{ij} = 1 \quad \forall j \in J \quad (20)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J. \quad (21)$$

5 Adjusting the layout

5.1 Setting options

All options are $\langle key \rangle = \langle value \rangle$ pairs. They can be set for the whole document or for one model:

```

% Preamble: all models of the document
\mpsetup[indent=3em, numsep=3em, expr_suffix={\;,}]

% One model only
\begin{mpformulation}[indent=1em]
...
\end{mpformulation}

```

Options given to the environment are local to it for the lengths and for `numbering`, `label`, `name_label`, `rel_align`, `quant_align`, `quant_overflow` and `quant_col`. The text and behaviour options (the suffixes, `st_text`, `st_indent`, `st_auto`, `st_inline`, `last_auto`, `keep_last` and the break modes) are global and remain in force after the environment.

5.2 Horizontal and vertical spacing

The horizontal layout of a line is:

|<- indent ->| expression |<- at least quant_gap ->| quantifier |<- numsep ->| (n)

`indent` (default 6em) is the space before the expression and the objective, `numsep` (default 2em) the space between the quantifier and the number. Vertically, `rowsep` (default 0.5\baselineskip) is added after each line, and `topsep` and `bottomsep` (default 0pt) before and after the environment.

```
\begin{mpformulation}[indent=1em, numsep=1em, rowsep=0pt, bottomsep=6pt]
  \constraint[MODEL7:constr1]{\sum_{i \in I} x_{ij} = 1}{\forall j \in J}
  \constraint[MODEL7:constr2]{x_{ij} \leq y_i}{\forall i \in I, \, j \in J}
\end{mpformulation}
Text after the model.
```

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J \quad (22)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J. \quad (23)$$

Text after the model.

5.3 Aligning on the relation

With `rel_align`, the constraints of an environment are aligned on their relation symbol ($=$, $\leq$, $\geq$, $\in$, $\dots$). The package splits each expression at its first relation outside braces; to choose another point, put `&` before the relation. Constraints whose expression contains `\\` are not aligned. The environment is measured before it is typeset, so the alignment is right on the first L^AT_EX run.

```
\begin{mpformulation}[rel_align]
  \constraint[MODEL18:constr1]{\sum_{i \in I} x_{ij} = 1}{\forall j \in J}
  \constraint[MODEL18:constr2]{x_{ij} \leq y_i}{\forall i \in I, \, j \in J}
  \constraint[MODEL18:constr3]{t_{i,j+1} - t_{ij} & \geq H^{\min}}{\forall i \in I, \, j \in J}
  \constraint[MODEL18:constr4]{y_i \in \{0, 1\}}{\forall i \in I}
\end{mpformulation}
```

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J \quad (24)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J \quad (25)$$

$$t_{i,j+1} - t_{ij} \geq H^{\min} \quad \forall i \in I, j \in J \quad (26)$$

$$y_i \in \{0, 1\} \quad \forall i \in I. \quad (27)$$

5.4 Left-aligned quantifiers

By default the quantifiers are flushed right. With `quant_align=left`, they start at a common column instead, just after the widest expression of the environment. An expression that reaches the column has its quantifier on the next line, still at the column. A quantifier too wide for the room left after the column is written unbroken on the next line, starting under the expression (indented by `expr_cont_indent`); it is wrapped at its commas only if it does not fit there either. With `quant_overflow=wrap`, it is wrapped at the column instead, each line starting at the column.

```
\begin{mpformulation}[quant_align=left, indent=3em, numsep=1em]
  \constraint[MODEL19:constr1]{\sum_{i \in I} x_{ij} = 1}{\forall j \in J}
  \constraint[MODEL19:constr2]{x_{ij} \leq y_i}{\forall i \in I, \, j \in J}
  \constraint[MODEL19:constr3]{t_{i+1,s} - t_{is} \geq H^{\min}}{\forall i \in I, \, s \in S, \, i < n}
  \constraint[MODEL19:constr4]{y_i \in \{0, 1\}}{\forall i \in I}
\end{mpformulation}
```

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J \quad (28)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J \quad (29)$$

$$t_{i+1,s} - t_{is} \geq H^{\min} \quad \forall i \in I, s \in S, i < n \quad (30)$$

$$y_i \in \{0, 1\} \quad \forall i \in I. \quad (31)$$

The column is computed from the widest expression of the environment, which is measured before it is typeset: the alignment is right on the first $\text{\LaTeX}$ run. The column is never placed beyond `quant_col_max` (default `0.6\linewidth`), so that one long expression does not squeeze all quantifiers; `quant_col` sets the column explicitly, as a distance from the left edge of the text.

In a narrow column, the default keeps each quantifier on one line:

```

\begin{mpformulation}[quant_align=left,
  indent=1em, numsep=0.5em]
\constraint[MODEL20:constr1]{x_{ij} \le
y_i}{\forall i \in I, j \in J}
\constraint[MODEL20:constr2]{t_{i+1,s} -
t_{is} \geq H}{\forall i \in I, s \in S, i < n, k \in K}
\constraint[MODEL20:constr3]{\sum_{s \in S_p} (1 - h_s) + \sum_{s \in \bar{S}_p} h_s \geq 1}{\forall p \in \Pi, b \in B}
\end{mpformulation}

```

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J \quad (32)$$

$$t_{i+1,s} - t_{is} \geq H \quad \forall i \in I, s \in S, i < n, k \in K \quad (33)$$

$$\sum_{s \in S_p} (1 - h_s) + \sum_{s \in \bar{S}_p} h_s \geq 1 \quad \forall p \in \Pi, b \in B. \quad (34)$$

The same model with `quant_overflow=wrap`:

```

\begin{mpformulation}[quant_align=left,
  indent=1em, numsep=0.5em, quant_
  overflow=wrap]
\constraint[MODEL21:constr1]{x_{ij} \le
y_i}{\forall i \in I, j \in J}
\constraint[MODEL21:constr2]{t_{i+1,s} -
t_{is} \geq H}{\forall i \in I, s \in S, i < n, k \in K}
\constraint[MODEL21:constr3]{\sum_{s \in S_p} (1 - h_s) + \sum_{s \in \bar{S}_p} h_s \geq 1}{\forall p \in \Pi, b \in B}
\end{mpformulation}

```

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J \quad (35)$$

$$t_{i+1,s} - t_{is} \geq H \quad \forall i \in I, s \in S, i < n, k \in K \quad (36)$$

$$\sum_{s \in S_p} (1 - h_s) + \sum_{s \in \bar{S}_p} h_s \geq 1 \quad \forall p \in \Pi, b \in B. \quad (37)$$

6 Long constraints and narrow columns

A model that fits a one-column page often does not fit a two-column journal. `mpformulation` measures every line and, when it is too wide, tries the following layouts in order, keeping the first that fits:

1. **Shrink the indent.** The line stays on one row; the indent is reduced as much as needed, down to `min_indent` (default 0pt).
2. **Break the quantifier.** The quantifier is split over several lines at its commas, next to the expression.
3. **Move the quantifier below.** The expression stays alone on the first line; the quantifier goes right-aligned on the line below.
4. **Break the expression.** Only if the expression itself is too wide and contains a break point `\\` written by the author.

The equation number is always on the last line, as with the `tbtags` option of `mathtools`. The lines of one constraint are kept on the same page.

The examples of this section are typeset in a column of about 7 cm.

6.1 Shrinking the indent

```
\begin{mpformulation}[indent=3em, numsep
=0.5em]
\constraint[MODEL8:constr1]{x_{ij} \leq
y_i}{\forall i \in I, \, j \in J}
\constraint[MODEL8:constr2]{\sum_{j \in
J} d_j x_{ij} \leq Q_i y_i}{\forall i
\in I}
\end{mpformulation}
```

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J \quad (38)$$

$$\sum_{j \in J} d_j x_{ij} \leq Q_i y_i \quad \forall i \in I. \quad (39)$$

The second constraint does not fit with the 3em indent, so it starts further left. Set `min_indent` equal to `indent` to forbid this.

6.2 Breaking the quantifier automatically

A quantifier that does not fit is broken at its commas. Lines are filled from left to right, each broken line ends with its comma, and spacing commands such as `\,` at the start of a new line are dropped. Commas inside `{}`, `()`, `[]`, `\{\}` and `\langle\rangle` are never break points, so (i, j) and $\{1, 2, 3\}$ stay together.

```
\begin{mpformulation}[indent=1em, numsep
=0.5em]
\constraint[MODEL9:constr1]{\sum_{k \in
K} x_{ijk} \leq 1}{\forall (i, j) \in
A, \, t \in \{1, \dots, T\}, \, s \in S}
\end{mpformulation}
```

$$\sum_{k \in K} x_{ijk} \leq 1 \quad \forall (i, j) \in A, \quad t \in \{1, \dots, T\}, s \in S. \quad (40)$$

6.3 Choosing the break yourself

To decide where a quantifier breaks, put `\\` there. The break is *optional*: it is used only if the quantifier does not fit on one line, otherwise `\\` is replaced by a space. With `quant_break=always` every `\\` is honored.

```

\begin{mpformulation}[indent=1em, numsep
=0.5em]
\constraint[MODEL10:constr1]{y_i \le 1}{
\forall i \in I, \forall \forall \forall t \in T}
\constraint[MODEL10:constr2]{h_{bd} - h
_{b,d-1} \geq z_{qd}}{\forall b \in B,
\, q \in Q^b, \forall d \in D}
\end{mpformulation}

```

$$y_i \leq 1 \quad \forall i \in I, \forall t \in T \quad (41)$$

$$h_{bd} - h_{b,d-1} \geq z_{qd} \quad \forall b \in B, q \in Q^b, d \in D. \quad (42)$$

The first constraint fits, so its $\forall$ is ignored. In the second, the break is taken at $\forall$ rather than at the first comma.

6.4 Moving the quantifier below

When the expression leaves too little room even for a broken quantifier, the quantifier moves to the next line, flushed right.

```

\begin{mpformulation}[indent=1em, numsep
=0.5em]
\constraint[MODEL11:constr1]{\sum_{s \in
S_p} (1 - h_s) + \sum_{s \in \bar{S}_p}
h_s + \sum_{q \in Q} w_q \geq 1}{
\forall p \in \Pi, \forall b \in B}
\end{mpformulation}

```

$$\sum_{s \in S_p} (1 - h_s) + \sum_{s \in \bar{S}_p} h_s + \sum_{q \in Q} w_q \geq 1 \quad \forall p \in \Pi, b \in B. \quad (43)$$

6.5 Breaking the expression

A relation wider than the column cannot be broken automatically: good break points depend on the meaning of the formula. Mark one with $\forall$ in the expression. It is used only as a last resort, and the continuation is indented by `expr_cont_indent` (default 2em). Start the continuation with $\{ \} +$ or $\{ \} \geq$ so that the operator keeps its spacing.

```

\begin{mpformulation}[indent=1em, numsep
=0.5em]
\constraint[MODEL12:constr1]{\sum_{s \in
S_p} (1 - h_s) + \sum_{s \in \bar{S}_p}
h_s \, \forall \, \{ \} + \sum_{q \in Q} w_q +
\sum_{r \in R} u_r \geq 1}{\forall p \in \Pi, \forall b \in B}
\end{mpformulation}

```

$$\sum_{s \in S_p} (1 - h_s) + \sum_{s \in \bar{S}_p} h_s + \sum_{q \in Q} w_q + \sum_{r \in R} u_r \geq 1 \quad \forall p \in \Pi, b \in B. \quad (44)$$

The quantifier is placed on the last expression line if it fits there (broken at its commas if needed), otherwise below. With `expr_break=always`, every $\forall$ in the expression is honored even when the line would fit.

6.6 Fine-tuning

quant_gap (default 1em) Minimum space between expression and quantifier. A smaller value keeps more quantifiers on the expression line.

min_indent (default 0pt) Smallest indent a too-wide line may use.

quant_break auto (default): optional `\` and automatic breaks at commas; **manual**: optional `\` only; **always**: every `\` is a break, no automatic break.

expr_break auto (default) or always.

If no layout fits (an expression wider than the column without `\`, or a quantifier piece without a comma), the line overflows into the margin and L^AT_EX reports an overfull `\hbox`.

7 Models across pages

A long model may run over several pages (or columns): the page can break between any two constraints, and the numbering continues. The package prevents the breaks that a typesetter would reject:

- never after the objective or after “subject to”: the objective, “subject to” and the first constraint are always on the same page;
- never inside a constraint set on several lines;
- never before the last constraint, so that it is not left alone at the top of a page (`keep_last=false` allows it).

To force a break between two constraints, write `\pagebreak` between them.

8 A complete example

The following model schedules trains that run each section of a line in an economic (*E*) or fast (*R*) mode. The layout options are given to the environment so that the example is self-contained; in a paper, put them once in the preamble with `\mpsetup`.

```
\begin{mpformulation}[rowsep=0.8\baselineskip, numsep=3em, indent=3em, bottomsep=10pt,
  expr_suffix={\;,}]
\objective[MODEL13:objective]{TT}{Minimize}{\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}
  e_{jk}x_{ijk} + \lambda\sum_{i\in I}L_i}
\constraint[MODEL13:constr1]{\sum_{k\in K} x_{ijk} = 1}{\forall i \in I, \, j \in J}
\constraint[MODEL13:constr2]{t_{i,j+1} = t_{ij} + \sum_{k\in K} r_{jk}x_{ijk}}{
  \forall i \in I, \, j \in J}
\constraint[MODEL13:constr3]{t_{i+1,s} - t_{is} \geq H^{\min}}{\forall i \in I, \, s
  \in S, \, i < n}
```

```

\constraint[MODEL13:constr4]{t_{i+1,s} - t_{is} \leq H^{\max}}{\forall i \in I, s \in S, i < n}
\constraint[MODEL13:constr5]{t_{i1} \geq \bar{t}_i - \Delta}{\forall i \in I}
\constraint[MODEL13:constr6]{t_{i1} \leq \bar{t}_i + \Delta}{\forall i \in I}
\constraint[MODEL13:constr7]{L_i \geq t_{im} - \bar{a}_i}{\forall i \in I}
\constraint[MODEL13:constr8]{\sum_{j \in J} x_{ijR} \leq R^{\max}}{\forall i \in I}
\constraint[MODEL13:constr9]{x_{ijE} = 0}{\forall i \in I^P, j \in J}
\constraint[MODEL13:constr10]{x_{ijE} + x_{i,j+1,R} \leq 1}{\forall i \in I, j \in J, j < m-1}
\constraint[MODEL13:constr11]{x_{ijR} + x_{i,j+1,E} \leq 1}{\forall i \in I, j \in J, j < m-1}
\constraint[MODEL13:constr12]{t_{i,j^*+1} \leq B + M(1-q_i)}{\forall i \in I}
\constraint[MODEL13:constr13]{t_{ij^*} \geq C - Mq_i}{\forall i \in I}
\end{mpformulation}
The objective~\eqref{MODEL13:objective} is minimized subject to
constraints~\labelcref{MODEL13:constr1}--\labelcref{MODEL13:constr13}.

```

$$(TT) \quad \text{Minimize} \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} e_{jk} x_{ijk} + \lambda \sum_{i \in I} L_i \quad (45)$$

subject to

$$\sum_{k \in K} x_{ijk} = 1, \quad \forall i \in I, j \in J \quad (46)$$

$$t_{i,j+1} = t_{ij} + \sum_{k \in K} r_{jk} x_{ijk}, \quad \forall i \in I, j \in J \quad (47)$$

$$t_{i+1,s} - t_{is} \geq H^{\min}, \quad \forall i \in I, s \in S, i < n \quad (48)$$

$$t_{i+1,s} - t_{is} \leq H^{\max}, \quad \forall i \in I, s \in S, i < n \quad (49)$$

$$t_{i1} \geq \bar{t}_i - \Delta, \quad \forall i \in I \quad (50)$$

$$t_{i1} \leq \bar{t}_i + \Delta, \quad \forall i \in I \quad (51)$$

$$L_i \geq t_{im} - \bar{a}_i, \quad \forall i \in I \quad (52)$$

$$\sum_{j \in J} x_{ijR} \leq R^{\max}, \quad \forall i \in I \quad (53)$$

$$x_{ijE} = 0, \quad \forall i \in I^P, j \in J \quad (54)$$

$$x_{ijE} + x_{i,j+1,R} \leq 1, \quad \forall i \in I, j \in J, j < m-1 \quad (55)$$

$$x_{ijR} + x_{i,j+1,E} \leq 1, \quad \forall i \in I, j \in J, j < m-1 \quad (56)$$

$$t_{i,j^*+1} \leq B + M(1-q_i), \quad \forall i \in I \quad (57)$$

$$t_{ij^*} \geq C - Mq_i, \quad \forall i \in I. \quad (58)$$

The objective (45) is minimized subject to constraints (46)–(58).

9 Reference

9.1 Commands and environment

`\begin{mpformulation}[\langle options \rangle] ... \end{mpformulation}`

Applies $\langle options \rangle$, adds `topsep` before and `bottomsep` after the block. The commands below also work outside the environment. The former name `constraints` is accepted for the environment.

`\constraint[\langle label \rangle]{\langle expression \rangle}{\langle quantifier \rangle}`

One constraint. $\langle expression \rangle$ in display style followed by the expression suffix; $\langle quantifier \rangle$ in text style followed by the quantifier suffix. Numbered and labelled if $\langle label \rangle$ is not empty. $\backslash\backslash$ in $\langle expression \rangle$ or $\langle quantifier \rangle$ is an optional break. Inside the environment, a `\constraint` directly followed by `\end` takes the last-line suffix.

`\lastconstraint[\langle label \rangle]{\langle expression \rangle}{\langle quantifier \rangle}`

As `\constraint`, with the last-line suffix. Optional inside the environment, where the last constraint is detected (see `last_auto`). Alias: `\constraintlast`.

`\objective[\langle label \rangle]{\langle name \rangle}{\langle sense \rangle}{\langle expression \rangle}`

Objective line at the constraints indent: formatted $\langle name \rangle$ (omitted if empty), $\langle sense \rangle$ in text, $\langle expression \rangle$ in display style. Numbered if $\langle label \rangle$ is not empty. $\backslash\backslash$ in $\langle expression \rangle$ is an optional break (see `obj_break`).

`\subjectto`

The “subject to” line. Optional: it is inserted automatically before the first constraint after `\objective` (see `st_auto`).

`\mpsetup{\langle options \rangle}`

Sets options for the rest of the document (or the current group). Alias: `\constraintsetup`.

9.2 Options

Key	Type	Default	Meaning
<code>indent</code>	length	<code>6em</code>	Indent of the expression and of the objective
<code>numsep</code>	length	<code>2em</code>	Space between quantifier and equation number
<code>rowsep</code>	length	<code>0.5\baselineskip</code>	Vertical space after each line
<code>topsep</code>	length	<code>0pt</code>	Space before the environment
<code>bottomsep</code>	length	<code>0pt</code>	Space after the environment
<code>expr_suffix</code>	text	empty	Appended to every expression (in math mode)
<code>quant_suffix</code>	text	empty	Appended to the quantifier of <code>\constraint</code>
<code>last_suffix</code>	text	<code>.</code>	Appended to the quantifier of the last constraint
<code>last_auto</code>	boolean	<code>true</code>	Detect the last constraint before <code>\end{mpformulation}</code>
<code>name_sep</code>	length	<code>1em</code>	Space after the model name
<code>sense_sep</code>	length	<code>1em</code>	Space after the sense
<code>st_text</code>	text	<code>subject to</code>	Text of <code>\subjectto</code>
<code>st_indent</code>	length	<code>\parindent</code>	Indent of <code>\subjectto</code> from the text edge, evaluated when used
<code>st_auto</code>	boolean	<code>true</code>	Insert “subject to” automatically after <code>\objective</code>
<code>quant_gap</code>	length	<code>1em</code>	Minimum space between expression and quantifier
<code>min_indent</code>	length	<code>0pt</code>	Smallest indent of a too-wide line
<code>quant_break</code>	choice	<code>auto</code>	<code>auto</code> , <code>manual</code> or <code>always</code> (Section 6)
<code>expr_break</code>	choice	<code>auto</code>	<code>auto</code> or <code>always</code>
<code>expr_cont_indent</code>	length	<code>2em</code>	Extra indent of expression continuation lines
<code>obj_break</code>	choice	<code>auto</code>	<code>auto</code> or <code>always</code> : use of <code>\\</code> in the objective
<code>numbering</code>	choice	<code>line</code>	<code>line</code> , <code>sub</code> (1a, 1b, ...) or <code>model</code> (one number)
<code>label</code>	text	empty	Label of the model number (environment only)
<code>name_label</code>	text	empty	Label of the model name (environment only)
<code>st_inline</code>	boolean	<code>false</code>	“subject to” on the first constraint line
<code>rel_align</code>	boolean	<code>false</code>	Align the constraints on their relation symbol
<code>keep_last</code>	boolean	<code>true</code>	No page break before the last constraint
<code>quant_align</code>	choice	<code>right</code>	<code>right</code> (flushed right) or <code>left</code> (common column, wrapped)
<code>quant_overflow</code>	choice	<code>below</code>	Left-aligned quantifier too wide after the column: <code>below</code> (unbroken on the next line) or <code>wrap</code> (wrapped at the column)
<code>quant_col</code>	length	<code>auto</code>	Column of left-aligned quantifiers (environment only)
<code>quant_col_max</code>	length	<code>0.6\linewidth</code>	Largest automatic column, evaluated when used

9.3 Macros and lengths

The options are stored in the following lengths and macros, which may also be changed directly with `\setlength` or `\renewcommand`.

Option	Stored in	Option	Stored in
<code>indent</code>	<code>\constraintLeftmargin</code>	<code>expr_suffix</code>	<code>\constraintExprSuffix</code>
<code>numsep</code>	<code>\constraintEqspace</code>	<code>quant_suffix</code>	<code>\constraintQuantSuffix</code>
<code>rowsep</code>	<code>\constraintSep</code>	<code>last_suffix</code>	<code>\constraintLastSuffix</code>
<code>topsep</code>	<code>\constraintTopsep</code>	<code>st_text</code>	<code>\constraintSubjecttoText</code>
<code>bottomsep</code>	<code>\constraintBottomsep</code>	<code>st_indent</code>	<code>\constraintSubjecttoIndent</code>
<code>name_sep</code>	<code>\constraintNameSep</code>	<code>quant_gap</code>	<code>\constraintQuantGap</code>
<code>sense_sep</code>	<code>\constraintSenseSep</code>	<code>min_indent</code>	<code>\constraintMinIndent</code>
<code>expr_cont_indent</code>	<code>\constraintExprContIndent</code>		

The model name is formatted by `\constraintNameFormat{\langle name \rangle}`, which defaults to `(#1)`.

9.4 Former option names

The package was developed under the name `constraints` up to version 1.12. Documents written for earlier versions keep working once they load `mpformulation`: the `constraints` environment and `\constraintsetup` remain available, and the former option names are accepted as aliases. The former macros `\constraintLHSSuffix`, `\constraintRHSSuffix`, `\constraintRHSlastsuffix` are still set by the options, and `\constraintRHSgap`, `\constraintLHScontIndent` are the same lengths as `\constraintQuantGap`, `\constraintExprContIndent`.

Former name	Current name
<code>sep</code>	<code>rowsep</code>
<code>eqspace</code>	<code>numsep</code>
<code>leftmargin</code>	<code>indent</code>
<code>suffix_left, suffix, lhs_suffix</code>	<code>expr_suffix</code>
<code>suffix_right, rhs_suffix</code>	<code>quant_suffix</code>
<code>rhs_gap</code>	<code>quant_gap</code>
<code>rhs_break</code>	<code>quant_break</code>
<code>lhs_break</code>	<code>expr_break</code>
<code>lhs_cont_indent</code>	<code>expr_cont_indent</code>
<code>suffix_right_last</code>	<code>last_suffix</code>
<code>subjectto_indent</code>	<code>st_indent</code>
<code>subjectto_text</code>	<code>st_text</code>

10 Limitations

- The model is not a display environment: `\abovedisplayskip` is not used (use `topsep`).
- Relation alignment (`rel_align`) does not apply to constraints whose expression contains `\\`.
- The body of the environment is read as an argument: verbatim commands such as `\verb` cannot be used inside it. With `rel_align` or `quant_align=left`, the body is typeset twice (a measuring pass, then the real one), so commands with global side effects written between the constraints are executed twice.
- Equation numbers are always on the right; the `leqno` option is ignored.

- Automatic breaks happen only at commas of the quantifier. A comma that is not a separator must be braced (`x_{i,j}`).
- `\objective` is broken only at the `\` written by the author.
- `numbering=sub` and `numbering=model` number the whole environment; a group of constraints inside a model cannot be sub-numbered separately.
- The text options (suffixes, `st_text`, `st_indent`, `last_auto`, `st_auto`, `st_inline`, `keep_last`, break modes) are global, even when given to the environment; `numbering`, `label`, `name_label` and `rel_align` are local.
- The default `numsep` is 2em in both one- and two-column documents; set it explicitly for narrow columns.
- The package requires L^AT_EX 2021-06-01 or later.

11 Change history

- 2.0 (2026/09/28)** Package renamed `mpformulation` (formerly `constraints`), as it now formats whole models; environment `mpformulation` and command `\mpsetup`. The former package name, environment name and `\constraintsetup` still work.
- 1.12 (2026/09/28)** Left-aligned quantifiers too wide for the room after the column go unbroken below the expression instead of being wrapped; option `quant_overflow`.
- 1.11 (2026/09/28)** Relation alignment and left-aligned quantifiers are right on the first run: the environment is measured before it is typeset (no `.aux` file and no second run needed).
- 1.10 (2026/09/28)** Left-aligned quantifiers at a common column, wrapped when needed: options `quant_align`, `quant_col`, `quant_col_max`.
- 1.9 (2026/09/28)** Options and macros renamed after the two parts of a constraint, the expression and the quantifier (for example `expr_suffix` and `quant_suffix` instead of `lhs_suffix` and `rhs_suffix`); see [Section 9](#). Former names kept as aliases.
- 1.8 (2026/09/28)** Model numbering (`numbering`, `label`, `name_label`); objective broken at `\` (`obj_break`); `st_inline`; relation alignment (`rel_align`); no page break after the objective and “subject to”, nor before the last constraint (`keep_last`); suffix attached to the LHS when the RHS is empty.
- 1.7 (2026/09/28)** Last constraint detected automatically; option `last_auto`.
- 1.6 (2026/09/28)** “subject to” inserted automatically after `\objective`; option `st_auto`.
- 1.5 (2026/09/28)** Layout fallback for long constraints: shrinking indent, RHS below the LHS, optional `\` in the LHS; equation number on the last line. Options `min_indent`, `lhs_break`, `lhs_cont_indent`.
- 1.4 (2026/09/28)** Automatic breaking of the RHS at its commas; `rhs_break>manual`.
- 1.3 (2026/09/28)** `\` in the RHS becomes an optional break; options `rhs_gap`, `rhs_break`.

1.2 (2026/09/28) Shorter option names; former names kept as aliases.

1.1 (2026/09/28) `\objective` and `\subjectto`.

1.0 (2025/09/07) First version.

License

© 2025–2026 FYP. This work may be distributed and/or modified under the conditions of the L^AT_EX Project Public License, either version 1.3 of this license or (at your option) any later version.