New MIP modeling constructs in Xpress Mosel to handle logical relations and certain nonlinear constraints

Susanne Heipcke[1], Yves Colombani[1], Sébastien Lannez[1]

Xpress Optimization, FICO, FICO House, Starley Way, Birmingham B37 7GN, UK

{SusanneHeipcke,YvesColombani,SebastienLannez}@fico.com

Mots-clés : *Mathematical modeling, Mixed-Integer Programming, linearization*

1 Introduction

The formulation of mathematical optimization models often involves certain nonlinear constructs, such as absolute values or logical relations. A widely used approach to handle such cases consists in linearizing the nonlinear expressions to make the model suitable for solving via a MIP solver, which may come at the expense of solution speed or quality.

Since the introduction of Special Ordered Sets (SOS), semi-continuous and partial integer variables in the late 1980s [1] an increasing number of special modeling constructs have been introduced into MIP solvers. In this talk we are going to take a look at a set of recent additions to the FICO Xpress Optimizer known as *General Constraints*, including a discussion of formulation alternatives via binary variables or indicator constraints, and their use in model formulation with Xpress Mosel.

2 General Constraints

General constraints are specific nonlinear expressions that are recognized by MIP solvers :

— absolute value, minimum or maximum value of discrete or continuous decision variables

— logical constraints : 'and' and 'or' over binary variables

— piecewise linear : can be used in place of SOS-2 formulations

The solver chooses the most appropriate representation and applies presolving to these constructs. Structural information that is available on the modeling level can thus be passed directly to the solver instead of having to apply transformations during matrix generation (e.g. the handling of piecewise linear expressions in AMPL [2]).

2.1 Absolute Value, Minimum Value, Maximum Value Expressions

Absolute value, minimum or maximum value expressions involving decision variables can be formulated by introducing additional binary variables and a few additional constraints. Using formulations based on indicator constraints removes the need for the model developer to come up with safe bound values that can introduce scaling issue when selected too large. The direct formulation via general constraints is more easily readable/maintainable and allows the solver to decide on the best suitable handling. We compare formulation options for $v = |x_1 - x_2|$ and $u = \max(x_1, ..., x_N)$ using Xpress Mosel (see example in Figure 1).

2.2 Piecewise Linear Expressions

Piecewise linear expressions can be employed to model a number of different situations :

Incremental pricebreaks : when buying a certain number of items discounts are applied

```
declarations
 x1, x2, v, d: mpvar
end-declarations
d is_binary
x1 <= U;  x2 <= U    ! Upper bound is needed
v >= x1 - x2;  v >= x2 - x1
v <= x1 - x2 + 2*U*d;  v <= x2 - x1 + 2*U*(1-d)
```

```
declarations
 x1, x2, v: mpvar
end-declarations
d is_binary
v >= x1 - x2;  v >= x2 - x1
indicator(-1, d, v <= x1 - x2)
indicator(1, d, v <= x2 - x1)
```

```
declarations
 x1, x2, v: mpvar
end-declarations

v = abs(x1 - x2)
```

FIG. 1 – MIP formulation alternatives for $v = |x_1 - x_2|$

incrementally, resulting in a piecewise linear continuous cost curve that can be represented via Special Ordered Sets of type 2 (SOS2), binary variables, or as a piecewise linear expression.
All items discount : when buying a certain number of items discounts are applied to *all* items, typically resulting in a discontinuous cost curve with piecewise linear segments that can be formulated via Special Ordered Sets of type 1 (SOS1), binary variables, or as a piecewise linear expression.
Approximating univariate non-linear functions : non-linear functions can be approximated by piecewise linear functions, represented via SOS2 or a piecewise linear expression.

2.3 Logical Constraints

For more than 10 years it has been possible to employ in a Mosel model formulation the logical constraints 'and', 'or', 'not', 'implies' and 'xor' that are transformed into indicator constraints on the modeling level, thus resulting in a MIP problem. With the introduction of support for logical constraints 'and' and 'or' on the solver level, a new decision variable type *boolvar* has been introduced for the formulation of logical constraints 'and', 'or', 'not' in the Mosel language that result in general constraints on binary variables for problem input to the Xpress MIP optimizer.

Many optimization problems that involve 'only if – then' conditions or 'either – or' choices can be expressed in a concise and more natural way with the help of logical constraints.

3 Concluding remarks

The introduction of new high-level modeling constructs for the formulation of MIP models eases the work of model developers. At the same time the solver receives more structural information that can be exploited in the solving process.

Xpress Mosel is free software [4], it can be downloaded from https://content.fico.com/xpress-optimization-community-license as part of the Xpress Community Edition (free of charge) that gives access to all solver functionality described in this article. Examples of the functionality discussed in this paper implemented with Xpress Mosel or the Xpress Python API can be found under https://examples.xpress.fico.com/example.pl.

Références

[1] Ashford, R.W. and Daniel, R.C. Some Lessons in Solving Practical Integer Problems. *J. Opl. Res. Soc.*, 45 (5) : 425–433, 1992.

[2] Fourer, R., Gay, D., Kernighan, B.W. *AMPL : A Modeling Language for Mathematical Programming*. The Scientific Press, San Francisco, 1993.

[3] FICO Xpress Optimization. *MIP formulations and linearizations*. Quick reference guide. http://www.fico.com/fico-xpress-optimization/docs/latest, 2009.

[4] Heipcke, S. and Colombani, Y. *Xpress Mosel : Modeling and Programming Features for Optimization Projects*. In : Neufeld, J. S. et al. (eds.) Operations Research Proceedings 2019, Springer, 677–683, 2020.