The AIMMS Presolver

Marcel Hunting
AIMMS Optimization Specialist

Webinar, July 15, 2015

Overview

> Introduction

> Preprocessing techniques

- Basic
- Advanced

> Activating the AIMMS presolver

> Analyse infeasible problems

AIMMS

Why Preprocessing?

- Speed up the solving process
- Get better solutions
- Improve numerical stability
- Detect infeasibilities

AIMMS

Why should AIMMS do Preprocessing?

> Not needed for linear problems because the linear solvers (CPLEX, Gurobi, CBC) do very advanced preprocessing

> But useful for nonlinear problems because many nonlinear solvers do no or very basic preprocessing

- Exceptions: KNITRO, BARON

> Optimization algorithms implemented in AIMMS can benefit from it

- Outer Approximation (MINLP)
- Multi-Start (NLP)

AIMMS

Basic Preprocessing Techniques

- Delete simple constraints
  - Constraint $x \leq 100$ becomes a bound
- Delete fixed variables
- Remove doubletons
  - Constraint $x = 2\,y$
- Delete duplicate constraints
- Delete redundant constraints
  - Constraint $\text{sqrt}(x) + y \leq 20$ can be removed if $x \leq 100$ and $y \leq 5$

AIMMS

Advanced Preprocessing Techniques

> Bounds tightening (feasibility based)
  - Variable *x*: range [0,inf) → range [10,55]
  - Linear & nonlinear constraints

> Linearize quadratic constraints
  - Multiplication of a binary variable with a bounded variable

> Improve coefficients
  - For models with binary variables

AIMMS

Feasibility Based Bound Tightening

Constraint:

$$L \le x + f(y_1,\ldots,y_n) \le U$$

Assume for some l_f and u_f:

$$l_f \le f(y_1,\ldots,y_n) \le u_f$$

Then:

$$x \ge L - u_f$$

$$x \le U - l_f$$

AIMMS

FBBT: Example

x: range [10,100]

y: range [5,8]

z: range [1,2]

Constraint: $x + 4\,y - 2\,z^3 = 60$

$x = 60 - 4\,y + 2\,z^3 \leq 60 - 20 + 16 = 56$

$x = 60 - 4\,y + 2\,z^3 \geq 60 - 32 + 2 = 30$

→ x: range [30,56]

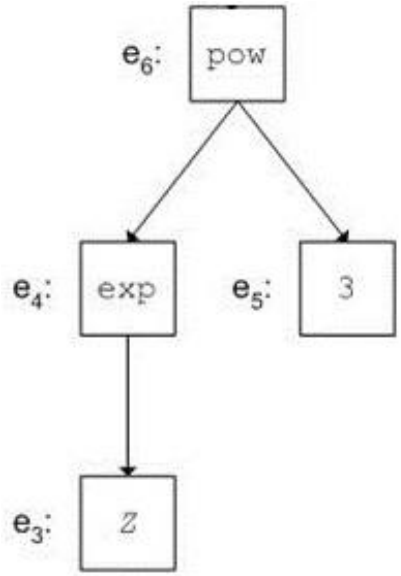
AIMMS

FBBT: Expression Tree

$(e^z)^3 \in [0,8]$

tighten bounds variables

$e^z \in [0,2]$

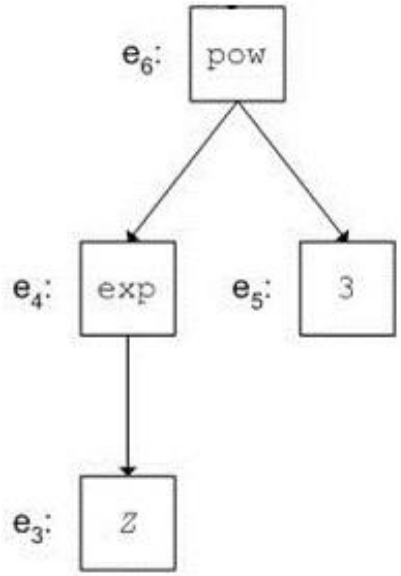


$z \in (-\text{inf}, \ln(2)]$

AIMMS

FBBT: Expression Tree

$(e^z)^3 \in [8,64]$



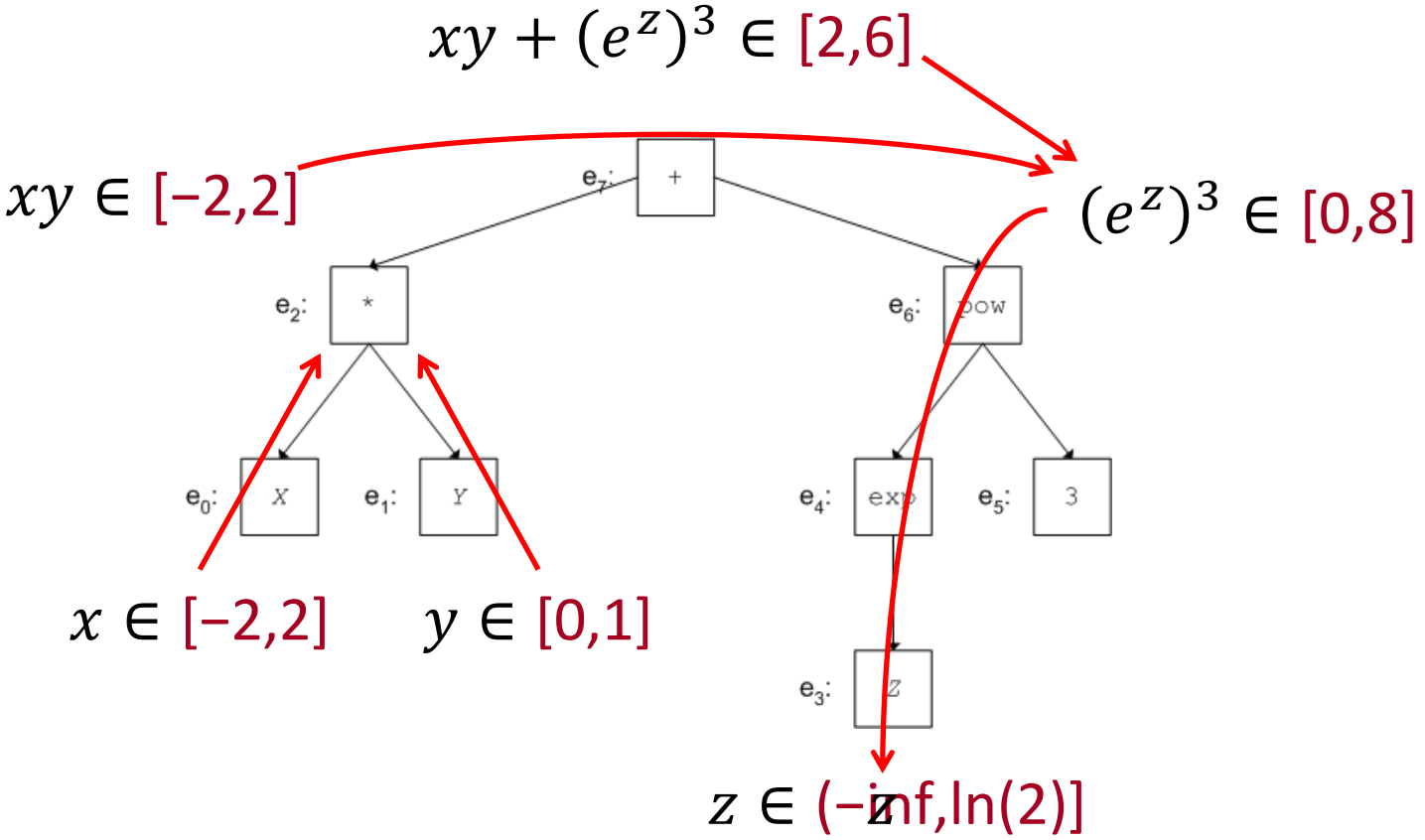
$e^z \in [2,4]$

tighten bounds expression

$z \in [\ln(2),\ln(4)]$

AIMMS

FBBT: Binary Operator



AIMMS

Linearize Constraints

Constraint

$$z_t \geq x_t y_t$$

with

x_t binary, $y_t \in [20,100]$, z_t free

$x_t = 1 \rightarrow z_t \geq y_t$

$x_t = 0 \rightarrow z_t \geq 0$

$$z_t \geq y_t + 100\,(x_t - 1)$$
$$z_t \geq 20\,x_t$$

AIMMS

Linearize Constraints (cont'd)

Constraint

$$(1 - x_t)\, y_t \le y_{t-1}$$

with

x_t binary, $y_t \in [0,100]$

$x_t = 1 \rightarrow y_{t-1} \ge 0$

$x_t = 0 \rightarrow y_t \le y_{t-1}$

$$y_t \le y_{t-1} + 100\, x_t$$

AIMMS

Improving Coefficients

$y + 12\,x \le 20$

x binary

$0 \le y \le 14$

$x = 0 \rightarrow y \le 20$ loose

$x = 1 \rightarrow y \le 8$ tight

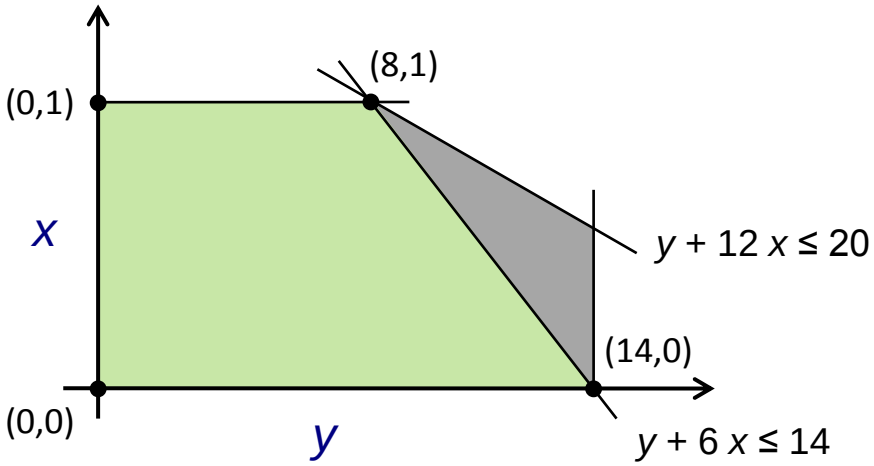
$y + 6\,x \le 14$

x binary

$0 \le y \le 14$

$x = 0 \rightarrow y \le 14$ tight

$x = 1 \rightarrow y \le 8$ tight



AIMMS

Improving Coefficients: Nonlinear

$$g(y_1,\ldots,y_k) + Mx \le b$$
$$x \text{ binary}$$

If we have an upper bound g^u on g such that $g^u < b$ then reformulate:

$$g(y_1,\ldots,y_k) + (M - b + g^u)\,x \le g^u$$
$$x \text{ binary}$$

AIMMS

Improving Coefficients: Probing

$g(y_1,\ldots,y_k) + Mx \le b$

x binary

$y \ge 0$

$y_i \le m_i x \quad i = 1,\ldots,k$

$x = 0 \rightarrow y_1 = \ldots = y_k = 0$ (implication)

If $g(0,\ldots,0) < b \rightarrow$ tighten

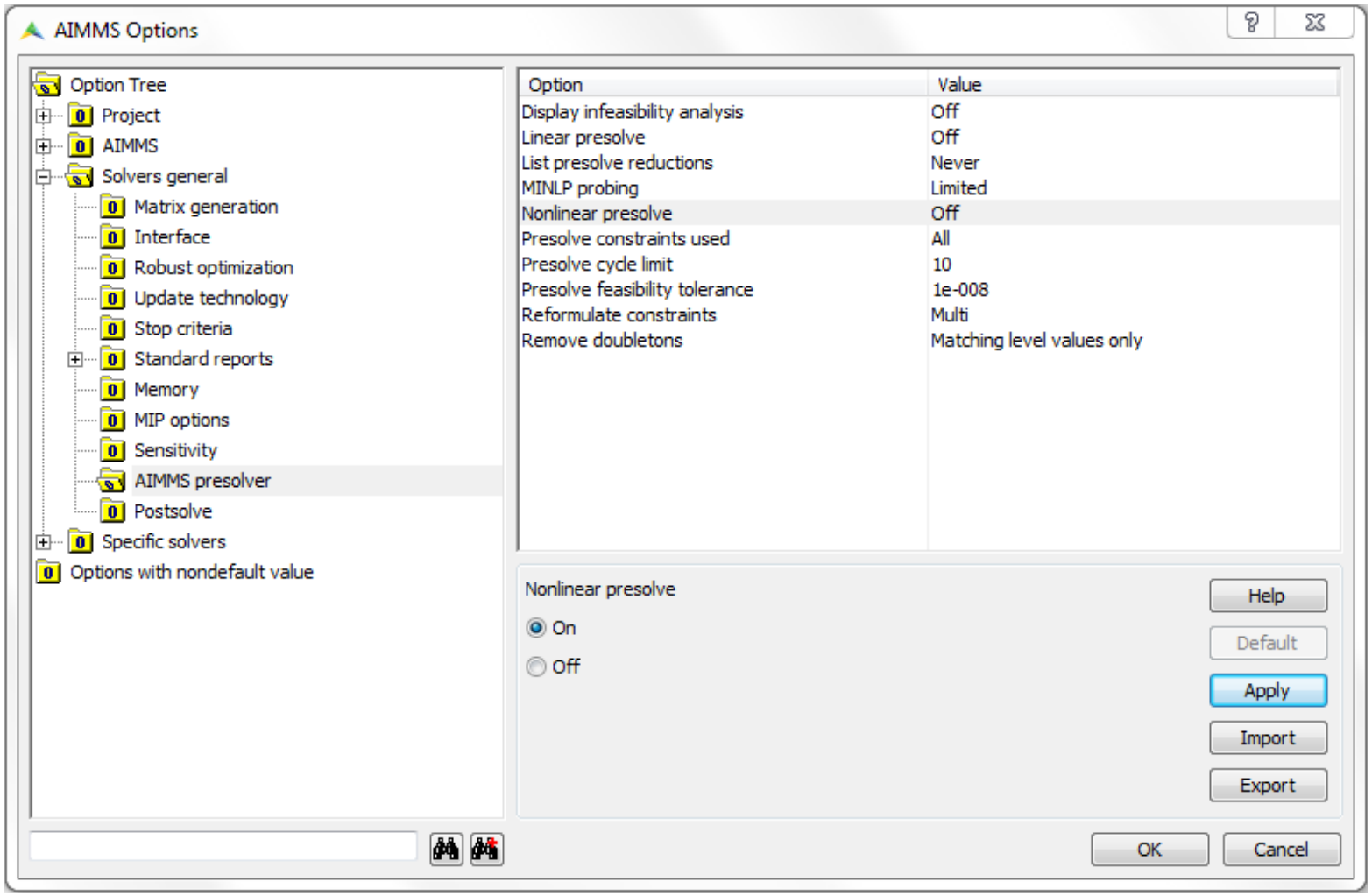
AIMMS

AIMMS Presolver Algorithm

```
RemoveDuplicateConstraints;

RemoveDoubletonVariables;

LinearizeConstraints;

while ( iter <= iterLimit and someBoundTightened ) do
    TightenVariableBounds;
    RemoveRedundantConstraints;
endwhile;

RemoveDoubletonVariables;

RemoveFixedVariables;

LinearizeConstraints;

ImproveCoefficients;
```


AIMMS

Activate Presolver: Normal Solve



AIMMS

Activate Presolver: GMP Solve

```
myGMP := GMP::Instance::Generate(MathProgram);
GMP::Instance::Solve(myGMP);
```

> Option 1: Set ‘Nonlinear presolve’ option (same as before)

> Option 2:

```
myGMP := GMP::Instance::Generate(MathProgram);
myPresolvedGMP := GMP::Instance::CreatePresolved(myGMP);
GMP::Instance::Solve(myPresolvedGMP);
```


AIMMS

Activate Presolver: Standard Algorithms

> Outer Approximation (MINLP):

```
myGMP := GMP::Instance::Generate(MathProgram);
GMPOuterApprox::UseNonlinearPresolver := 1;
GMPOuterApprox::DoOuterApproximation(myGMP);
```

> Multi-Start (NLP):

```
myGMP := GMP::Instance::Generate(MathProgram);
MulStart::UsePresolver := 1;
MulStart::DoMultiStart(myGMP,…);
```


AIMMS

Infeasibility Analysis

> Activated by setting options ‘Display Infeasibility Analysis’ and ‘Nonlinear Presolve’

> Warnings :

- Local nonlinear solver (CONOPT, KNITRO, SNOPT, IPOPT, AOA) might return Locally Infeasible while problem is feasible
- For an infeasible problem the AIMMS presolver often cannot detect that the problem is indeed infeasible

AIMMS

References

> Language Reference

- The AIMMS Presolver: Chapter 17.1 (AIMMS 4.8)

> Gondzio, J., Presolve analysis of linear programs prior to applying the interior-point method, *INFORMS Journal on Computing* **9**, 1997, pp. 73-91.

> Savelsbergh, M.W.P., Preprocessing and Probing Techniques for Mixed Integer Programming Problems, *ORSA Journal on Computing* **6**, 1994, pp. 445-454.

AIMMS