AIMMS Function Reference - GMPLinearization Procedures and Functions

GMP::Linearization Procedures and Functions

AIMMS supports the following procedures and functions for creating and managing linearizations associated with a generated mathematical program instance:

- GMP::Linearization::Add
- GMP::Linearization::AddSingle
- GMP::Linearization::Delete
- GMP::Linearization::GetDeviation
- GMP::Linearization::GetDeviationBound
- GMP::Linearization::GetLagrangeMultiplier
- GMP::Linearization::GetType
- GMP::Linearization::GetWeight
- GMP::Linearization::RemoveDeviation
- GMP::Linearization::SetDeviationBound
- GMP::Linearization::SetType
- GMP::Linearization::SetWeight

GMP::Linearization::Add

The procedure `GMP::Linearization::Add` adds a linearization row to a generated mathematical program (*GMP1*) with respect to a solution (column level values and row marginals) of a second generated mathematical program (*GMP2*) for each row in a set of nonlinear constraints of that second generated mathematical program.

```
GMP::Linearization::Add(
     GMP1,                 ! (input) a generated mathematical program
     GMP2,                 ! (input) a generated mathematical program
     solution,             ! (input) a solution
     constraintSet,        ! (input) a set of nonlinear constraints
     deviationsPermitted,  ! (input) a binary parameter
     penaltyMultipliers,   ! (input) a double parameter
     linNo,                ! (input) a linearization number
     [jacTol]              ! (optional) the Jacobian tolerance
     )
```

Arguments:

GMP1
: An element in `AllGeneratedMathematicalPrograms`.

GMP2
: An element in `AllGeneratedMathematicalPrograms`.

solution
: An integer scalar reference to a solution in the solution repository of *GMP2*.

constraintSet
: A subset of `AllNonLinearConstraints`.

deviationsPermitted
: A binary parameter over `AllNonLinearConstraints` indicating whether deviations are permitted for these linearizations. If yes, a new column will also be added to *GMP1* with an objective coefficient equal to the double value given in *penaltyMultiplier* multiplied with the row marginal of the row in *solution*.

penaltyMultipliers
: A double parameter over `AllNonLinearConstraints` representing the penalty multiplier that will be used if *deviationsPermitted* indicates that a deviation is permitted for the linearization.

linNo
: An integer scalar reference to the rows and columns of the linearization.

jacTol
: The Jacobian tolerance; if the Jacobian value (in absolute sense) of a

column in a nonlinear row is smaller than this value, the column will not be added to the linearization of that row. The default is 1e-5.

Return value:

The procedure returns 1 on success, or 0 otherwise.

Remarks:

- This procedure fails if one of the constraints is ranged.
- Rows and columns added before for *linNo* will be deleted first.
- This procedure will fail if *GMP2* contains a column that is not part of *GMP1*. A column that is part of *GMP1* but not of *GMP2* will be ignored, i.e., no coefficient for that column will be added to the linearizations.
- The formula for the linearization of a scalar nonlinear inequality $g(x, y) \leq b_j$ around the point $(x, y) = (x^0, y^0)$ is as follows.

$$g(x^0, y^0) + \nabla g(x^0, y^0)^T \begin{bmatrix} x - x^0 \\ y - y^0 \end{bmatrix} - z_j \leq b_j$$

  where $z_j \geq 0$ is the extra column that is added if a deviation is permitted.
- For a scalar nonlinear equality $g(x, y) = b_j$ the sense of the linearization depends on the shadow price of the equality in the *solution*. The sense will be '≤' if the shadow price is negative and the optimization direction is minimization, or if the shadow price is positive and the optimization direction is maximization. The sense will be '≥' if the shadow price is positive and the optimization direction is minimization, or if the shadow price is negative and the optimization direction is maximization.
- By using the suffices `.ExtendedConstraint` and `.ExtendedVariable` it is possible to refer to the rows and columns that are added by `GMP::Linearization::Add`:
  - Constraint `c.ExtendedConstraint('Linearization`*k*`',j)` is added for each nonlinear constraint `c(j)`.
  - Variable `c.ExtendedVariable('Linearization`*k*`',j)` is added for each nonlinear constraint `c(j)` if a deviation is permitted for constraint `c(j)`.

  Here *k* denotes the value of the argument *linNo*.

See also:

The routines `GMP::Linearization::AddSingle` and `GMP::Linearization::Delete`. See Section 21.3.6 of the Language Reference for more details on extended suffices.

GMP::Linearization::AddSingle

The procedure `GMP::Linearization::AddSingle` adds a single linearization row to a generated mathematical program (*GMP1*) with respect to a solution (column level values and row marginals) of a second generated mathematical program (*GMP2*).

```
GMP::Linearization::AddSingle(
     GMP1,                  ! (input) a generated mathematical program
     GMP2,                  ! (input) a generated mathematical program
     solution,              ! (input) a solution
     row,                   ! (input) a scalar reference
     deviationPermitted,    ! (input) a binary scalar
     penaltyMultiplier,     ! (input) a double scalar
     linNo,                 ! (input) a linearization number
     [jacTol]               ! (optional) the Jacobian tolerance
     )
```

Arguments:

GMP1
: An element in `AllGeneratedMathematicalPrograms`.

GMP2
: An element in `AllGeneratedMathematicalPrograms`.

solution
: An integer scalar reference to a solution in the solution repository of *GMP2*.

row
: A scalar reference to an existing nonlinear row in *GMP2* for which the linearization is added to *GMP1*.

deviationPermitted
: A binary scalar indicating whether a deviation is permitted for this linearization. If yes, a new column will also be added to *GMP1* with an objective coefficient equal to the double value given in *penaltyMultiplier* multiplied with the row marginal of the row in *solution*.

penaltyMultiplier
: A double value representing the penalty multiplier that will be used if *deviationPermitted* indicates that a deviation is permitted for the linearization.

linNo
: An integer scalar reference to the rows and columns of the linearization.

jacTol
: The Jacobian tolerance; if the Jacobian value (in absolute sense) of a column in *row* is smaller than this value, the column will not be added to the linearization. The default is 1e-5.

Return value:

The procedure returns 1 on success, or 0 otherwise.

Remarks:

- This procedure fails if the row is ranged.
- Rows and columns added before for *linNo* will be deleted first.
- This procedure will fail if *GMP2* contains a column that is not part of *GMP1*. A column that is part of *GMP1* but not of *GMP2* will be ignored, i.e., no coefficient for that column will be added to the linearizations.
- The formula for the linearization of a scalar nonlinear inequality $g(x, y) \leq b_j$ around the point $(x, y) = (x^0, y^0)$ is as follows:

$$g(x^0, y^0) + \nabla g(x^0, y^0)^T \begin{bmatrix} x - x^0 \\ y - y^0 \end{bmatrix} - z_j \leq b_j$$

  where $z_j \geq 0$ is the extra column that is added if a deviation is permitted.
- For a scalar nonlinear equality $g(x, y) = b_j$ the sense of the linearization depends on the shadow price of the equality in the *solution*. The sense will be '≤' if the shadow price is negative and the optimization direction is minimization, or if the shadow price is positive and the optimization direction is maximization. The sense will be '≥' if the shadow price is positive and the optimization direction is minimization, or if the shadow price is negative and the optimization direction is maximization.
- By using the suffices `.ExtendedConstraint` and `.ExtendedVariable` it is possible to refer to the row and column that are added by `GMP::Linearization::AddSingle`:
  - Constraint `c.ExtendedConstraint('Linearization`*k*`',j)` is added for row `c(j)`.
  - Variable `c.ExtendedVariable('Linearization`*k*`',j)` is added for row `c(j)` if a deviation is permitted.

  Here *k* denotes the value of the argument *linNo*.

Examples:

Assume that 'prod03' is a mathematical program with the following declaration (in aim format):

```
VARIABLE:
   identifier :  i1
   range      :  {1..5} ;

VARIABLE:
   identifier :  i2
   range      :  {1..5} ;

VARIABLE:
   identifier :  objvar ;

CONSTRAINT:
   identifier :  e1
```

```
   definition :  - 3*i1 - 2*i2 + objvar = 0 ;

CONSTRAINT:
   identifier :  e2
   definition :  - i1*i2 <= -3.5 ;

MATHEMATICAL PROGRAM:
   identifier :  prod03
   objective  :  objvar
   direction  :  minimize
   type       :  MINLP ;
```

Assume that AIMMS has executed the following code in which a mathematical program instance 'gmp1' is generated from 'prod03', its integer variables are relaxed, and it is solved.

```
gmp1 := GMP::Instance::Generate(prod03);
GMP::Instance::SetMathematicalProgrammingType(gmp1,'RMINLP');
GMP::Instance::Solve(gmp1);
```

The optimal solution is `i1` = 1.528 and `i2` = 2.291, with Jacobian values −2.291 and −1.528 for `i1` and `i2` respectively. This solution is stored at position 1 in the solution repository of 'gmp1'. If we have a second generated mathematical program 'gmp2' with the same variables as 'gmp1' then

```
GMP::Linearization::AddSingle(gmp2,gmp1,1,e2,0,0,1);
```

will add a row

```
e2.ExtendedConstraint('Linearization1'):
   - 2.291 * i1 - 1.528 * i2 <= -7 ;
```

to 'gmp2'.

See also:

The routines `GMP::Linearization::Add` and `GMP::Linearization::Delete`. See Section 21.3.6 of the Language Reference for more details on extended suffices.

GMP::Linearization::Delete

The procedure `GMP::Linearization::Delete` deletes a set of rows and columns corresponding to a linearization in a generated mathematical program.

```
GMP::Linearization::Delete(
     GMP,         ! (input) a generated mathematical program
     linNo        ! (input) a linearization number
     )
```

Arguments:

GMP
: An element in `AllGeneratedMathematicalPrograms`.

linNo
: An integer scalar reference to the rows and columns of the linearization.

Return value:

The procedure returns 1 on success, or 0 otherwise.

See also:

The routines `GMP::Linearization::Add` and `GMP::Linearization::AddSingle`.

GMP::Linearization::GetDeviation

The function GMP::Linearization::GetDeviation returns the deviation of a linearization of a row in a generated mathematical program.

```
GMP::Linearization::GetDeviation(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo   ! (input) a linearization number
     )
```

Arguments:

GMP
: An element in AllGeneratedMathematicalPrograms.

row
: A scalar reference to an existing nonlinear row in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

Return value:

The function returns the deviation of the row.

See also:

The routines GMP::Linearization::SetDeviationBound and GMP::Linearization::GetDeviationBound.

GMP::Linearization::GetDeviationBound

The function `GMP::Linearization::GetDeviationBound` returns the deviation bound of a linearization of a row in a generated mathematical program. The lower bound of the extra column generated for the linearization is always 0; this function returns the upper bound.

```
GMP::Linearization::GetDeviationBound(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo   ! (input) a linearization number
     )
```

Arguments:

GMP
: An element in `AllGeneratedMathematicalPrograms`.

row
: A scalar reference to an existing nonlinear row in in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

Return value:

The function returns the deviation upperbound of a linearization.

See also:

The routines `GMP::Linearization::SetDeviationBound` and `GMP::Linearization::GetDeviation`.

GMP::Linearization::GetLagrangeMultiplier

The function `GMP::Linearization::GetLagrangeMultiplier` returns the Lagrange multiplier used when adding the linearization of a row to a generated mathematical program. (In other words, the marginal value of the row that was used when the linearization was added.)

```
GMP::Linearization::GetLagrangeMultiplier(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo   ! (input) a linearization number
     )
```

Arguments:

GMP
: An element in `AllGeneratedMathematicalPrograms`.

row
: A scalar reference to an existing nonlinear row in in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

Return value:

The function returns the Lagrange multiplier used when adding the linearization of a row.

See also:

The procedure `GMP::Linearization::Add`.

GMP::Linearization::GetType

The function GMP::Linearization::GetType returns the row type of a linearization of a row in a generated mathematical program.

```
GMP::Linearization::GetType(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo   ! (input) a linearization number
     )
```

Arguments:

GMP
: An element in AllGeneratedMathematicalPrograms.

row
: A scalar reference to an existing nonlinear row in in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

Return value:

An element in the set AllRowTypes.

See also:

The procedure GMP::Linearization::SetType.

GMP::Linearization::GetWeight

The function GMP::Linearization::GetWeight returns the weight of a linearization of a row in a generated mathematical program. The weight of a linearization is defined as the objective coefficient of the column that was added to the generated mathematical program when the linearization was added and if a deviation was permitted.

```
GMP::Linearization::GetWeight(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo   ! (input) a linearization number
     )
```

Arguments:

GMP
: An element in AllGeneratedMathematicalPrograms.

row
: A scalar reference to an existing nonlinear row in in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

Return value:

The function returns the weight of the linearization.

Remarks:

- This function returns 0 if no extra column was added for the linearization.
- If the objective coefficient of the deviation column (if any) was not changed, the weight equals the penalty multiplier multiplied with the marginal value of the row that was used when the linearization was added with GMP::Linearization::Add or GMP::Linearization::AddSingle.

See also:

The procedures GMP::Linearization::Add, GMP::Linearization::AddSingle and GMP::Linearization::SetWeight.

GMP::Linearization::RemoveDeviation

The procedure GMP::Linearization::RemoveDeviation removes the deviation of a linearization of a row in a generated mathematical program. That is, it deletes the extra column created (if any) when adding the linearization of the row to the generated mathematical program.

```
GMP::Linearization::RemoveDeviation(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo   ! (input) a linearization number
     )
```

Arguments:

GMP
: An element in AllGeneratedMathematicalPrograms.

row
: A scalar reference to an existing nonlinear row in in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

Return value:

The procedure returns 1 on success, or 0 otherwise.

See also:

The routines GMP::Linearization::GetDeviation, GMP::Linearization::Add and GMP::Linearization::AddSingle.

GMP::Linearization::SetDeviationBound

The procedure `GMP::Linearization::SetDeviationBound` sets the deviation bound of a linearization of a row in a generated mathematical program. The lower bound of the extra column generated for the linearization is always 0; this procedure sets the upper bound.

```
GMP::Linearization::SetDeviationBound(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo,  ! (input) a linearization number
     value   ! (input) a scalar value
     )
```

Arguments:

GMP
: An element in `AllGeneratedMathematicalPrograms`.

row
: A scalar reference to an existing nonlinear row in in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

value
: A scalar value representing the deviaton upper bound of the row.

Return value:

The procedure returns 1 on success, or 0 otherwise.

See also:

The routines `GMP::Linearization::GetDeviationBound`, `GMP::Linearization::GetDeviation` and `GMP::Linearization::RemoveDeviation`.

GMP::Linearization::SetType

The procedure `GMP::Linearization::SetType` sets the row type of linearization of a row in a generated mathematical program.

```
GMP::Linearization::SetType(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo,  ! (input) a linearization number
     rowtype ! (input) a row type
     )
```

Arguments:

GMP
: An element in `AllGeneratedMathematicalPrograms`.

row
: A scalar reference to an existing nonlinear row in in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

rowtype
: An element (or element parameter or element valued expression) in the predeclared set `AllRowTypes`.

Return value:

The procedure returns 1 on success, or 0 otherwise.

See also:

The function `GMP::Linearization::GetType`.

GMP::Linearization::SetWeight

The procedure `GMP::Linearization::SetWeight` sets the weight of a linearization of a row in a generated mathematical program. The weight of a linearization is defined as the objective coefficient of the column that was added to the generated mathematical program when the linearization was added and if a deviation was permitted.

```
GMP::Linearization::SetWeight(
     GMP,    ! (input) a generated mathematical program
     row,    ! (input) a scalar reference
     linNo,  ! (input) a linearization number
     value   ! (input) a scalar value
     )
```

Arguments:

GMP
: An element in `AllGeneratedMathematicalPrograms`.

row
: A scalar reference to an existing nonlinear row in in the matrix.

linNo
: An integer scalar reference to the rows and columns of the linearization.

value
: A scalar value representing the new weight of the row.

Return value:

The procedure returns 1 on success, or 0 otherwise.

See also:

The function `GMP::Linearization::GetWeight`.